Лабораторная работа №2
Разработка требований

Цель: выявлить и описать пользовательские требования в виде вариантов использования (Use Cases).

План занятия:

1. Изучить теоретические сведения.
2. Выполнить практическое задание по лабораторной работе.
3. Оформить отчёт и ответить на контрольные вопросы.

Теоретические сведения

Требование (Requirement) – описание того, какие функции и с соблюдением каких условий должен выполнять программный продукт в процессе решения полезной для пользователя задачи.

Значение требований:

– Позволяют понять, что и с соблюдением каких условий система должна делать.

– Предоставляют возможность оценить масштаб изменений и управлять изменениями.

– Являются основой для формирования плана проекта (в том числе плана тестирования).

– Помогают предотвращать или разрешать конфликтные ситуации.

– Упрощают расстановку приоритетов в наборе задач.

– Позволяют объективно оценить степень прогресса в разработке проекта.

Работа над требованиями включает следующие этапы: выявление требований, анализ требований (моделирование бизнес-процессов, прототипирование интерфейсов, приоритезация требований, результат этапа – визуализация требований), документирование требований (результат этапа – спецификация), тестирование (валидация) требований. Работу с требованиями на этапах выявления, анализа, документирования, как правило, выполняет бизнес- аналитик. Тестирование требований выполняет тестировщик.

В иерархии требований существует три уровня: уровень бизнес-требований, уровень пользовательских требований, уровень продуктных требований (функциональные и нефункциональные требования).

Бизнес-требования выражают цель, ради которой разрабатывается продукт (зачем он нужен, какая от него ожидается польза).

Пользовательские требования описывают задачи, которые пользователь может выполнять с помощью разрабатываемой системы, и по своей сути представляют собой недетализированные функциональные требования. Поскольку здесь уже появляется описание поведения системы, требования этого уровня могут быть использованы для оценки объёма работ, стоимости проекта, времени разработки. Пользовательские требования оформляются в виде вариантов использования (Use Cases), пользовательских историй (User Stories), пользовательских сценариев (User Scenarios).

Функциональные требования описывают поведение системы, т.е. её действия (вычисления, преобразования, проверки, обработку и т.д.). Нефункциональные требования описывают свойства системы (удобство использования, безопасность, надёжность, расширяемость и т.д.), которыми она должна обладать при реализации своего поведения.

Выявление и описание требований: Use Case.

Вариант использования (Use Case) продукта описывает последовательность взаимодействия системы и внешнего действующего лица. Действующим лицом может быть человек, другая система ПО или аппаратное устройство, взаимодействующее с системой для достижения некой цели.

Варианты использования меняют традиционный подход к сбору информации: пользователей не спрашивают, что с их точки зрения, должна делать система, а выясняют, какие задачи собирается с ее помощью решать пользователь. Цель такого подхода — описать все подобные задачи. До включения каждого варианта использования в утвержденную версию требований заинтересованные в проекте лица проверяют, не выходит ли он за границы проекта. Теоретически в конечный набор вариантов использования должна входить вся желаемая функциональность системы.

Описание варианта использования включает следующие категории:

- уникальный идентификатор;
- имя, кратко описывающее задачи пользователи в формате «глагол + объект», например «разместить заказ»;
- краткое текстовое описание на естественном языке;
- список предварительных условий, которые должны быть удовлетворены до начала разработки варианта использования;
- выходные условия, описывающие состояние системы после успешного завершения разработки варианта использования;
- пронумерованный список действий, иллюстрирующий

последовательность этапов взаимодействия лица и системы от предварительных условий до выходных условий.

Пример варианта использования приведен в таблице 2.1.

Таблица 2.1 – Пример варианта использования

Идентификатор и название:	UC-4 Запросить химикат
Основное действующее лицо:	Сотрудник, разместивший заказ на химикат
Описание:	Сотрудник, разместивший заказ на химикат, указывает в запросе необходимый химикат, вводя его название или идентификатор или импортируя его структуру из соответствующего графического средства. Система выполняет запрос, предлагая контейнер с химикатом со склада или позволяя создать запрос на заказ у поставщика.
Триггер:	Сотрудник указывает, что хочет заказать химикат.
Предварительные условия:	PRE-1. Личность пользователя аутентифицирована. PRE-2. Пользователь имеет право запрашивать химикаты. PRE-3. База данных по запасам химикатов в данный момент доступна.
Выходные условия:	POST-1. Запрос сохраняется в Chemical Tracking System. POST-2. Запрос отправлен на склад химикатов или поставщику.
Нормальное направление развития варианта использования:	4.0 Запросить химикат со склада 1. Сотрудник указывает требуемый химикат. 2. Система перечисляет контейнеры с необходимым химикатом, имеющиеся на складе. 3. Сотрудник может просмотреть историю любого контейнера. 4. Сотрудник выбирает определенный контейнер или просит отправить запрос поставщику (см. 4.1). 5. Сотрудник вводит остальную информацию, чтобы завершить запрос. 6. Система сохраняет запрос и отправляет его на склад химикатов.

Окончание таблицы 2.1

Альтернативное направление развития варианта использования:	4.1 Запросить химикат у поставщика 1. Сотрудник ищет химикат по каталогам поставщика (см. 4.1.E1). 2. Система отображает список поставщиков, где также указаны размеры, класс и цена контейнеров. 3. Сотрудник выбирает поставщика, размер, класс и количество контейнеров. 4. Сотрудник вводит остальную информацию, необходимую для запроса. 5. Система сохраняет запрос и перенаправляет его поставщику.
Исключения:	4.1.E1. Химиката нет в продаже 1. Система отображает сообщение «У поставщиков нет такого химиката». 2. Система предлагает сотруднику запросить другой химикат или выйти из программы. 3а. Сотрудник просит запросить другой химикат. 4а. Система заново начинает нормальное направление варианта использования. 3б. Сотрудник решает выйти из системы. 4б. Система завершает вариант использования.
Бизнес-правила:	BR-28, BR-31

Существует несколько сценариев варианта использования (см. таблицу 2.1).

Один сценарий считается нормальным направлением развития (normal course) варианта использования, его также называют основным направлением, главным успешным сценарием и благоприятным путем. Нормальное направление для варианта использования «Запрос химиката» — запрос химиката, который есть на складе.

Другие допустимые сценарии из варианта использования, называются альтернативными направлениями (alternative courses) или вторичными сценариями (secondary scenarios). Они также могут привести к успешному выполнению задания и удовлетворяют выходным условиям варианта использования. Однако они представляют вариации решения задачи или диалоговой последовательности, необходимой для выполнения задачи. В определенной точке принятия решений в диалоговой последовательности нормальное направление может перейти в альтернативное, а затем вернуться обратно в нормальное.

Условия, препятствующие успешному завершению задания, называются исключениями (exceptions). Если в процессе сбора информации не указано, как обрабатывать исключение, то возможны два пути: 1) разработчики предложат лучший по их мнению способ обработки исключений; 2) при генерации пользователем неверного условия произойдет сбой системы, так как никто не предусмотрел такой ситуации. Иногда исключения рассматриваются как тип альтернативного направления, однако эти понятия следует разделять. Не обязательно реализовывать каждое альтернативное направление, которое определяют для варианта использования; кроме того, можно отложить его реализацию до следующего выпуска. Однако необходимо реализовать исключения, из-за которых завершение сценариев может оказаться неуспешным.

Расширение (extend) и включение (include).

При составлении вариантов использования часто можно столкнуться с ситуацией, когда альтернативное направление варианта использования само по себе можно выделить в автономный вариант использования. В таком случае можно расширить (extention) нормальное направление, включив этот отдельный вариант использования в нормальный поток.

Пример: Вариант использования "Запросить химикат" может включать в себя поиск по каталогу поставщика. Но при этом запросить химикат можно и без поиска по каталогам, а поиск по каталогу может выполняться как отдельная бизнес-задача пользователей. Поэтому логично расширить вариант использования "Запросить химикат" отдельным вариантом использования "Поиск по каталогам поставщика". Такая связь будет означать, что при выполнении варианта использования "Запросить химикат" может выполняться, а может и не выполняться вариант использования "Поиск по каталогам поставщика.

Иногда же несколько вариантов использования имеют общие наборы этапов. Чтобы избежать повторения этих этапов в каждом варианте использования, можно определить отдельный вариант использования и указать, что он включен (include) в другие варианты использования как подвариант.

Пример: Если покупатель совершает покупку товара, то он обязательно должен его оплатить. Нет случая, когда покупатель просто за что-то платит (т.е. оплата товара не является отдельной независимой деятельностью покупателя), но при этом процесс оплаты сам по себе достаточно сложный, включающий различные шаги и альтернативные варианты (оплата различными способами). Поэтому логично его выделить в отдельный вариант использования, при этом включить в вариант использования "Покупка товара". Такая связь будет означать,

что при выполнении варианта использования "Купить товар" обязательно будет задействован и вариант использования "Оплатить товар".

Определение вариантов использования.

Определить варианты использования можно несколькими способами:

– сначала определить действующие лица, а затем бизнес-процессы, в которых каждое лицо участвует;

– выразить бизнес-процессы в терминах определенных сценариев, обобщить сценарии в варианты использования и определить действующие лица для каждого варианта; определить внешние события, на которые система должна реагировать, а затем соотнести эти события с участвующими лицами и определенными вариантами использования;

– определить вероятные варианты использования на основе функциональных требований; если какие-либо требования невозможно проследить до какого-либо варианта использования, необходимо задуматься, нужны ли они.

Как правило, пользователи сначала определяют самые важные варианты использования, поэтому порядок предлагаемых тем позволит получить представление о приоритетах.

Преимущества применения вариантов использования состоят в том, что каждый вариант сосредоточен на поставленной задаче и пользователе. Тщательное изучение этапов взаимодействия лица и системы помогает еще на ранних стадиях разработки выявить неясности и неточности, а также позволяет составить варианты тестирования на основе вариантов использования. Способ с применением вариантов использования позволяет выявить функциональные требования, с помощью которых пользователи будут выполнять конкретные задачи. Кроме прочего варианты использования облегчают расстановку приоритетов требований. Высшим приоритетом обладают те функциональные требования, которые созданы на основе вариантов использования с высшим приоритетом. Высший приоритет назначается по следующим причинам:

– варианты использования описывают один из основных бизнес-процессов, активизируемых системой;

– многие пользователи часто обращаются к ним;

– их запросил привилегированный класс пользователей;

– они предоставляют возможности, необходимые для соответствия требованиям;

– функции других систем зависят от их наличия.

Существуют также и преимущества технического характера. С помощью варианта использования можно выявить некоторые важные объекты предметной

области и их взаимоотношения. Разработчики, использующие объектно-ориентированные методы проектирования, могут преобразовать варианты использования в объектные модели, такие, как диаграммы классов и диаграммы последовательностей.

Практическое задание:

1. Придумать идею и бизнес-цели подлежащего разработке программного продукта.
2. Определить действующие лица и сформулировать наиболее вероятные варианты использования подлежащего разработке программного продукта.
3. Полностью описать три варианта использования подлежащего разработке программного продукта.
4. Для каждого варианта использования указать уникальный идентификатор; имя в формате «глагол + объект»; краткое текстовое описание; предварительные условия; выходные условия; пронумерованный список действий нормального направления развития.
5. Для каждого варианта использования при необходимости указать пронумерованный список действий альтернативного направления (направлений) развития.
6. Для каждого варианта использования при необходимости указать исключения.
7. Оформить отчет и защитить лабораторную работу.

Содержание отчета:

1. Цель работы.
2. Описание вариантов использования подлежащего разработке программного продукта.
3. Выводы по работе.

Контрольные вопросы:

1. Что такое требование?
2. Какие значения имеют требования на проекте?
3. Какие существуют этапы работы над требованиями?
4. Кто выполняет работу с требованиями?
5. Какие существуют уровни требований?
6. Что такое вариант использования?
7. Для чего нужен вариант использования?
8. Какие элементы входят в состав описания варианта использования?
9. Что такое основной сценарий варианта использования?

10. Что такое альтернативный сценарий варианта использования?

11. Что описывают в исключениях варианта использования?

12. В чем отличие альтернативного сценария от исключения в описании варианта использования?

13. Какие существуют преимущества у вариантов использования как одного из способов описания требований?