

Лабораторная работа №1. Разработка и анализ требований к инфокоммуникационной системе

Цель работы

- изучить методику определения требований к инфокоммуникационной системе;
- разработать программные документы на создание системы, на основании которых будет организована реализация проекта.

Задание на выполнение лабораторной работы

Для выбранного варианта инфокоммуникационной системы

1. Определите набор требований на создание системы.
 - a. Сформируйте **видение (концепцию)** – понимание проекта в целом, бизнес-цели, которых хочет достичь заказчик от внедрения системы.

Концепция (видение) проекта содержит обобщенное представление долгосрочных целей и назначение нового продукта.

- b. Создайте **пользовательские истории** (user story).

Пользовательские истории составляются в свободной форме, в виде историй или некоторых сценариев использования системы. Каждая история имеет условного рассказчика (автора, пользователя) истории, повествующего о наиболее значимых для исполнения требований к проектируемой системе.

- c. Основываясь на пользовательских историях и бизнес-целях определите исходное концептуальное представление (концептуальную модель системы) в виде **диаграммы Use Case** (прецедентов, вариантов использования) в нотации UML и **текстового описания** (ниже в рекомендациях указано, что имеется в виду).
 - d. Определите основную функциональность (перечень функциональных требований), которую должна предоставлять система, представьте ее описание в виде **диаграммы требований** по нотации SysML [3].
2. Разработайте дополнительную спецификацию, словарь терминов и оформите **документ-концепцию** (“видение”) (из первого пункта задания) (см. п.7.1-7.7 стр. 108-122 [2]).
 3. Разработайте **спецификацию требований** к программному обеспечению (Modern Software Requirements Specification) согласно шаблону [4,5,6].

Методические рекомендации по разработке моделей

Рекомендации по разработке концепции программной системы

В **концепции** (видение) проекта обычно отражают сбалансированное описание требований, удовлетворяющих различным заинтересованным лиц, что в итоге определяет обобщенное представление долгосрочных целей и назначение нового продукта. В [1, глава 5, с. 91] описывается порядок создания документа концепции.

Составьте сжатое положение о концепции проекта, обобщающее долгосрочные цели и назначение нового продукта. В этом документе следует отразить сбалансированную концепцию, удовлетворяющую различных заинтересованных лиц. Она может быть несколько идеалистичной, но должна основываться на существующих или предполагаемых рыночных факторах, архитектуре предприятия, стратегическом направлении развития корпорации или ограничениях ресурсов.

Примените шаблон [1, с. 97], состоящий из ключевых слов, которые подходят для документа о концепции продукта (изучите пример формулировки концепции для приложения системы контроля химикатов):

• **Для** [целевая аудитория покупателей] • **Который** [положение о потребностях или возможностях] • **Эта (этот)** [имя продукта] • **Является** [категория продукта] • **Который(ая)** [основные функции, ключевое преимущество, основная причина для покупки или использования] • **В отличие от** [основной конкурирующий продукт, текущая система или текущий бизнес-процесс] • **Наш продукт** [положение об основном отличии и преимуществе нового продукта].

Рекомендации по выявлению пользователей программной системы

Класс пользователей является подмножеством пользователей продукта, которые являются подмножеством клиентов продукта, а те в свою очередь являются подмножеством заинтересованных лиц. Один человек может принадлежать разным классам пользователей, например, администратор иногда работает с системой как рядовой пользователь. Помимо всего прочего, пользователей продукта можно подразделять по характерным признакам и группировать их на отдельные классы. Изучите предлагаемые в [1, с.115] квалификационные различия и выделите отдельные классы т роли пользователей. На данном этапе рекомендуется обратить внимание на следующие признаки: по привилегиям доступа и уровню безопасности (например, рядовой пользователь, пользователь-гость, администратор), а также по задачам, которые им приходится решать при выполнении бизнес-операций.

Рекомендации по разработке модели программной системы

Пользовательская история – это короткое, простое описание функции с точки зрения человека (пользователь или клиент системы), которому нужна новая возможность. Пользовательские истории составляются в свободной форме, в виде историй или некоторых сценариев использования системы. Каждая история имеет условного рассказчика (автора, пользователя) истории, повествующего о наиболее

значимых для исполнения требований к проектируемой программной системе. Рекомендуется пользовательские истории составлять в соответствии с неким шаблоном, например, «Как *<тип пользователя>*, я хочу *<цель>*, чтобы *<причина>*» [1, с.168]. Эти истории являются основой программной модели, которая показывает функции системы (собственно варианты использования), их окружение (актеры) и связи (отношения) между прецедентами и актерами.

Диаграмма вариантов использования является исходным концептуальным представлением или концептуальной моделью системы в процессе ее проектирования и разработки. Вариант использования (use case) описывает последовательность взаимодействия системы и внешнего действующего лица, в результате которого действующее лицо получает полезный результат [1, с.171]. Класс пользователя (который не обязательно должен быть человеком) в пользовательской истории соответствует основному действующему лицу в варианте использования. Изучите примеры соответствия вариантов использования и пользовательских историй [1, с.168, таблица 8-2].

Разработка диаграммы вариантов использования преследует следующие цели:

- определить общие границы и контекст моделируемой предметной области на начальных этапах проектирования системы ([1] глава 5).
- сформулировать общие требования к функциональному поведению проектируемой системы ([1] глава 8).
- разработать исходную концептуальную модель системы для ее последующей детализации в форме логических и физических моделей ([1] главы 1-4).
- подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями ([1] глава 10).

Таким образом, диаграммы вариантов использования позволяют получить высокоуровневое визуальное представление о требованиях пользователей. Изучите пример диаграммы варианта использования в нотации UML [1, с.172, рис. 8-2].

На диаграммах вариантов использования каждое действующее лицо и вариант использования рекомендуется сопровождать описанием (сценарием), поскольку такие текстовые описания служат средством связи между лицами, часто не имеющими специальной подготовки. Поэтому простой текст обычно является наилучшим выбором. Описание варианта использования должно включать в себя пояснение, предусловие, потоки событий (основной и альтернативные, если таковые есть) и постусловие [1, с.175-180].

Изучите возможные варианты представления текстовых сценариев на основе предлагаемых шаблонов [1, с.174; 2, (вся книга об этом)], выберите наиболее подходящий для вашей задачи.

Составьте несколько вариантов использования для вашего текущего проекта на основании выбранного шаблона. Включите в них все альтернативные направления развития и исключения.

На основе вариантов использования следует сформулировать **функциональные требования**, необходимые для успешной реализации каждого варианта использования. В этом случае придется установить соответствие между вариантами использования и связанными с ними функциональными требованиями. Функциональные требования можно писать с точки зрения того, что делает система или что делает пользователь. Требования следует излагать последовательно, например «Система должна» или «Пользователь должен», затем — активный глагол, а после — наблюдаемый результат, также надо указать инициирующие условия или триггеры, вследствие которых система ведет себя определенным образом. Изучите общие шаблоны требований [1, с. 243] и характерные примеры удачных и неудачных формулировок [1, с. 244-258].

Функциональные требования к системе («рамки решения») и соответствующие им задачи разработчика («рамки проекта») требования удобно свести в таблицу (см. табл. 1).

Таблица 1.

ID	Описание «рамки решения»	ID	Описание «рамки проекта»
Функция: Название функции			
1	Система обеспечивает возможность пользователям.....	101	Создание меню.....
		102	Реализация просмотра результатов....

Проверьте варианты использования с клиентами (если их нет, то можно провести мыслительный эксперимент), чтобы убедиться в правильности определения модели программной системы, а также в том, что рассмотрены все вариации основного направления, учтены и обработаны все исключения, которые клиенты посчитали разумными.

Диаграмма требований применяется для визуального представления и организации требований к системе, она позволяет отображать взаимосвязи между различными требованиями (например, “родитель—потомок”), а также соотносить с другими элементами модели системы (компонентами, блоками, тестами).

Требования отображаются как блоки, соединенные линиями, которые указывают на их происхождение, зависимости и группировку. Эта диаграмма относится к отдельной категории в SysML, наряду со структурными и поведенческими диаграммами. Существуют специализированные инструменты для создания и работы с диаграммами требований SysML, например, [Visual Paradigm](#).

Отчетность по результатам занятия

По результатам работы необходимо оформить и защитить отчет.

В отчете должны быть приведены результаты выполнения заданий, разработанные диаграммы (для графического представления рекомендуется использовать результаты практического занятия №1).

В выводах укажите различия между языками UML и SysML и предложите практические рекомендации для выбора подходящего инструмента моделирования в зависимости от особенностей проектируемой системы.

Литература

1. Вигерс К. Разработка требований к программному обеспечению. 3-е изд., дополненное /Пер. с англ. — М.: Издательство «Русская редакция»; СПб.: БХВ-Петербург, 2014. — 736 с.
2. Ларман К. Применение UML и шаблонов проектирования. 2-е издание.: Пер. с англ. — М. : Издательский дом “Вильямс”, 2004. — 624 с.
3. [SysML](#)
4. Стандарт IEEE 830-1998 Методика составления спецификаций требований к программному обеспечению, рекомендуемая Институтом Инженеров по Электротехнике и Радиоэлектронике (IEEE) [Электронный ресурс] Режим доступа: IEEE-830-1998_RU.doc.
5. Шаблон SRS [Электронный ресурс] Режим доступа: rup_srs.dot.
6. Образец пакета Modern SRS Package [Электронный ресурс] Режим доступа: Образец пакета SRS.pdf.
7. Вигерс К. Жемчужины разработки. Чему мы научились за 50 лет создания ПО. — СПб.: Питер, 2024. — 368 с.