Вопрос 3. Анализ требований к программному обеспечению

1. Основные понятия

Анализ требований — часть процесса разработки программного обеспечения, включающая в себя:

- сбор требований к программному обеспечению (ПО),
- их систематизацию,
- выявление взаимосвязей,
- документирование.

В англоязычной среде также говорят о дисциплине «инженерия требований» (англ. Requirements Engineering). В процессе сбора требований важно принимать во внимание возможные противоречия требований различных заинтересованных лиц, таких как заказчики, разработчики или пользователи.

Полнота и качество анализа требований играют ключевую роль в успехе всего проекта. Требования к ПО должны быть документируемые, выполнимые, тестируемые, с уровнем детализации, достаточным для проектирования системы. Требования могут быть функциональными и нефункциональными.

Анализ требований включает три типа деятельности:

- **Сбор требований** — общение с клиентами и пользователями, чтобы определить, каковы их требования; анализ предметной области.
- **Анализ требований** — определение, являются ли собранные требования неясными, неполными, неоднозначными или противоречащими; решение этих проблем; выявление взаимосвязи требований.
- **Документирование требований** — может производиться в различных формах, таких как: простое описание, сценарии использования, пользовательские истории, или спецификации процессов.

Анализ требований может быть длинным и трудным процессом, во время которого вовлечены много тонких психологических навыков. Новые системы изменяют окружающую среду и отношения между людьми, поэтому важно определить все заинтересованные лица, принять во внимание все их потребности и гарантировать, что они понимают значения новых систем.

Аналитики могут использовать несколько методов, чтобы выявить следующие требования от клиента: проведение интервью, использование фокус-групп или создание списков требований. Более современные методы включают создание прототипов и сценариев использования. Где необходимо, аналитик будет использовать комбинацию этих методов, чтобы выявить точные требования заинтересованных лиц так, чтобы была создана система, которая удовлетворяет деловые потребности.

Процесс анализа требований к информационной системе включает следующие фазы:

- Разработка требований
- Выявление требований
- Анализ требований
- Спецификация требований
- Проверка требований
- Управление требованиями

Разработка программного средства начинается с этапа формулирования требований к ПС, на котором, исходя из пожеланий заказчика, формируется документ, определяющий задачи разработчиков, - **внешнее описание программного средства.**

Этот документ играет роль постановки задачи, содержит необходимую информацию по применению ПС и является исходным документом для процессов:

1. конструирования и кодирования программ, входящих в ПО;
2. разработки документации по применению ПО;
3. разработки комплекта тестов для тестирования ПО.

Исходным документом для разработки внешнего описания является определение требований к программному обеспечению. Через этот документ передается от заказчика к разработчику информация относительно требуемого содержания программ. Разработчик должен выполнить анализ области применения разрабатываемой системы с точки зрения определения требований к ней.

Требование это некое свойство программного обеспечения, которым должна обладать система или ее компонент, чтобы:

- решить проблемы пользователей при достижении поставленной цели;
- удовлетворить требования контракта, стандарта, спецификации либо иной формальной документации.

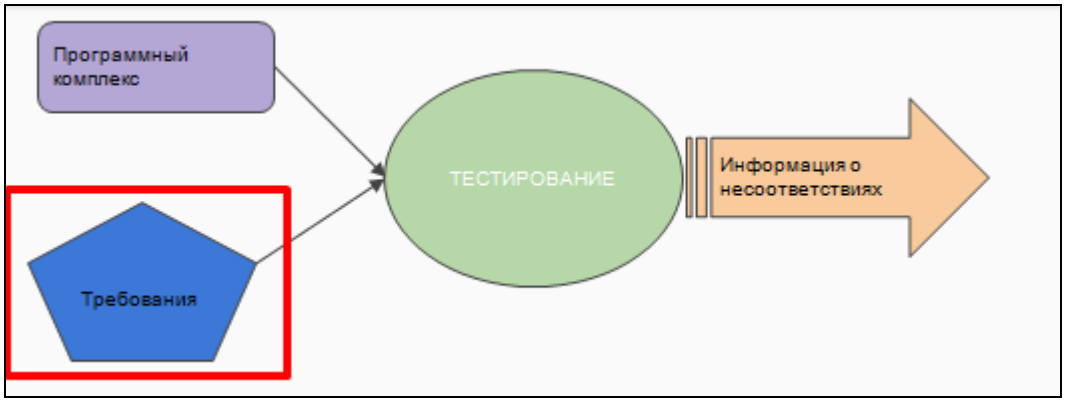
2. Определение требований к программному обеспечению

Технические требования к системе должны охватывать:

- функции и возможности системы;
- коммерческие и организационные требования;
- требования пользователя; требования безопасности и защиты; эргономические
- требования; требования к интерфейсам; эксплуатационные требования;
- требования к сопровождению;
- проектные ограничения и квалификационные требования.

Требования к системе должны быть оценены с учетом следующих критериев:

- соответствие потребностям заказчика;
- тестируемость;
- выполнимость проектирования системной архитектуры;
- возможность эксплуатации и сопровождения.



Виды и свойства требований

Требования к ПО определяют условия пользователей на внешнее его поведение и разработчиков на его параметры. Требования можно разбить на две большие группы – функциональные и нефункциональные.

Функциональные требования являются детальным описанием поведения и сервисов системы, ее функционала. Они определяют то, что система должна уметь делать. Сюда же относятся ограничения на данные и поведение системы. Спецификация функциональных требований включает в себя описание функций.

Нефункциональные требования не являются описанием функций системы. Они описывают такие характеристики системы, как надежность, особенности поставки (наличие инсталлятора, документации), уровень качества. Сюда же могут относиться требования на средства и процесс разработки системы, требования к переносимости, соответствию стандартам и т.д.

Нефункциональные требования определяют условия выполнения функций; принципы взаимодействия со средами или другими системами, учитывают время работы, защиту данных, а также стандарты качества для достижения отдельных показателей качества.

К ПС предъявляются следующие нефункциональные требования:

- к применению (качество пользовательского интерфейса, учебные курсы и др.);
- к производительности;
- к надежности и отказоустойчивости;
- к интерфейсным внешним атрибутам, с которыми взаимодействует система;
- к конфиденциальности, безопасности и защите данных;
- к одновременности доступа к системе пользователей и т. п.

Требования бывают:

- **Прямыми,** то есть формализованными в технической документации, спецификациях, User Story;
- **Косвенными —** проистекают из прямых, либо являются негласным стандартом для данной продукции, либо основаны на опыте и здравом смысле использования продукта или подобных ему.

Среди требований можно выделить:

Требования пользователей опираются на цели и задачи, которые позволит решать будущая система. К способам представления этого вида требований относятся варианты использования, сценарии, прецеденты, таблицы «событие-отклик» и т.п.

Системные требования определяют внешние условия для выполнения системных функций и ограничений на создаваемый продукт, а также требования к описанию подсистем. Системные требования накладывают ограничения на архитектуру системы, средства ее представления и функционирования.

Требования к атрибутам качества представляют собой некоторые ограничения к свойствам функций или к системе, важные для пользователей или разработчиков. Например, переносимость, целостность, устойчивость к сбоям.

Требования бизнеса:

1. Высокоуровневые цели организации или заказчика(Контекст)
2. Цели, создания системы и критерии их достижения.
3. Ключевые требования к решению и их приоритеты.

4. Список **стейкхолдеров** (Лица заинтересованные в системе)
5. Ограничения на решения

Например, *требования к приложению может состоять из:*
перечня бизнес- процессов, бизнес-правил и концептуальной модели предметной области.

Требования пользователей могут быть трех видов:
Сценарий пользователя

Например: *Терминал удостоверяется, что пополнение возможно, и запрашивает у Пользователя номер телефона и, если нужно, код оператора. Пользователь сообщает Терминалу запрошенные данные. Терминал удостоверяется, что данные введены корректно*

Пользовательские истории — способ описания требований, к разрабатываемой системе, сформулированный, как одно или более предложений на повседневном или деловом языке. Цель пользовательских историй состоит в том, чтобы быть в состоянии оперативно и без накладных затрат реагировать на быстро изменяющиеся требования реального мира.

Например: *как <Роль/Персона пользователя> я <Хочу что – то получить>, <С такой – то целью>*
или
как <Пользователь>, я <Хочу управлять рекламными объявлениями>, <Чтобы удалять устаревшие или ошибочные объявления>

Вариант использования — описание поведения системы, когда она взаимодействует с кем – то (или чем - то) из внешней среды. Система может отвечать на внешние запросы или сама выступать инициатором взаимодействия.

Например:

1. *Пользователь захотел разместить объявление.*
2. *Пользователь зашел в систему.*
3. *Пользователь авторизовался в системе.*
4. *Пользователь создал объявление.*
5. *Система отобразила сообщение об успешном создании объявления.*

Требования пользователей обычно не содержит деталей реализации и пишется на языке целей пользователей. Поэтому их удобно согласовывать с заказчиком как «Единицу поставки». Польза таких требований для разработчика в том, что он видит не отдельное «система должна…», а **контекст использования** той или иной функции. Какие функции будут выполнены, прежде чем будет вызвана эта? В каком виде и почему будут введены данные? Можно ли менять этот реализованный класс или это отразится на согласованных сценариях работы пользователя с системой?

Это понимание позволяет разработчику лучше планировать работу над реализацией отдельных объектов и функций, а также снимает часть вопросов о используемых форматах данных.

Требования пользователей являются отличной базой для формирования тестовых сценариев, так как они описывают в каком контексте должно производиться каждое действие пользователя. Это тестируемые требования по умолчанию, так как в них всегда указана цель, которой нужно достигнуть и какие шаги надо для этого воспроизвести.

Преимущества использования таких требований:
- Дают представление о поведении системы.
- Понятны заказчику и разработчикам.

- Позволяют описать множество альтернатив (исключений).
- Содержат в себе перечень функциональности системы.
- удобны для поддержки системы. Чтоб выявить ошибку, разобраться, на каком шаге что пошло не так.

Недостатки:

- Не обеспечивают полноту всех функциональных требований, если в систему должна быть заложена сложная бизнес-логика.
- Сценарии использования плохо подходят для документирования требований не основанных на взаимодействии с системой (таких как алгоритм или математические требования) или нефункциональных требований (такие как платформа, производительность, синхронизация, безопасность).
- Следование шаблонам не гарантирует качество сценариев. Качество зависит только от навыков создателя сценария.

Рекомендации по разработке требований пользователя:

- Основной сценарий не больше 3 – 9 шагов.
- Не включать элементы дизайна.
- Использование одного уровня детализации на всех шагах.
- Не использовать «Если»

Требования должны обладать следующими важными свойствами:

- Ясность, недвусмысленность - однозначность понимания требований заказчиком и разработчиками.
- Полнота и непротиворечивость.
- Необходимый уровень детализации. Это могут быть описание свойств предметной области или техническое задание.
- Прослеживаемость. Важно видеть то или иное требование в различных моделях, документах, в коде системы.
- Тестируемость и проверяемость.
- Модифицируемость. Определяет процедуры внесения изменений в требования.
- Единичность, атомарность.
- Последовательность, завершённость.
- Актуальность, обязательность.
- Выполнимость.

Цикл работы с требованиями

В современных информационных технологиях процесс жизненного цикла программного обеспечения, на котором фиксируются требования на разработку системы, является определяющим для задания функций, сроков и стоимости работ, а также показателей качества, которых необходимо достигнуть в процессе разработки.

Выдвижение требований проводится путем обсуждения проекта, анализа предметной области и определения подходов к проектированию промежуточных продуктов на этапах жизненного цикла.

Методы определения требований:

- Анкетирование
- Мозговой штурм
- Наблюдение за производственной деятельностью
- Анализ нормативной документации
- Анализ моделей деятельности

- Анализ конкурентных продуктов
- Анализ статистики использования предыдущих версий системы

Заказчик и разработчик совместно проводят обсуждение проблем проекта, сбор требований, их анализ, пересмотр, определение необходимых ограничений и документирование. Выделение требований нацелено на выявление всех возможных источников требований и ограничений на работу системы и извлечение требований из этих источников.

Анализ требований направлен на обнаружение и устранение противоречий и неоднозначностей в требованиях, их уточнении и систематизации.

Способы проверки требований:

- анализ,
- осмотр,
- демонстрация,
- тестирование.

Описание требований выполняется для их оформления в виде структурированного набора документов и моделей, которые могут анализироваться и оцениваться с разных позиций. В итоге этот набор документов должен быть утвержден как официальная формулировка требований к системе.

Валидация требований решает задачу оценки понятности сформулированных требований и их характеристик, необходимых для разработки ПО (в первую очередь, непротиворечивости и полноты, а также соответствия стандартам на техническую документацию).

Литература:

1. Анализ требований разработки ПО / https://unetway.com/tutorial/analiz-trebovanij-razrabotki-po

2. Кудлаев А.А. Лекции по технологии программирования: Основные процессы жизненного цикла программных средств. Учебное пособие. М., МИИГАиК, 2011, 71 стр. (стр. 19-21) / http://miigaik.ru/vtiaoai/tutorials/8.pdf

3. Дорофеев М. Требования к программному обеспечению и их анализ: онлайн-презентация / https://ppt-online.org/199986