Вопрос 2. Разработка технических спецификаций на программные компоненты и их взаимодействие

1. Варианты формализации требований

Источниками требований к проектированию приложений могут быть:

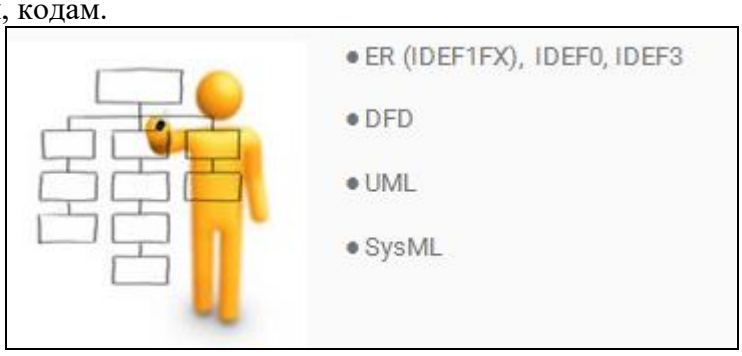
- Федеральное и муниципальное отраслевое законодательство (Конституция, законы, распоряжения)
- Нормативное обеспечение организации (Регламенты, положения, уставы, приказы)
- Текущая организация объекта автоматизации
- Модели деятельности (Диаграммы бизнес - процессов)
- Представления и ожидания потребителей и пользователей системы
- Опыт использования аналогичного ПО
- Конкурирующие программные продукты

Разработанные требования представляются в специальном документе, который является основой заключения контракта на разработку системы между заказчиком и разработчиком. Формализация требований в проекте может быть разной – это зависит от его величины, принятого процесса разработки, используемых инструментальных средств, а также тех задач, которые решают формализованные требования. Более того, может существовать параллельно несколько формализаций, решающих различные задачи:

Неформальная постановка требований в переписке по электронной почте. Хорошо работает в небольших проектах, при вовлеченности заказчика в разработку.

Требования в виде документа – описание предметной области и ее свойств, техническое задание как приложение к контракту, функциональная спецификация для разработчиков.

Требования в виде графа в одном из средств поддержки требований (IBM Rational RequisitePro, DOORS, Borland CaliberRM и др.). Такое представление требований удобно при их частом изменении, при отслеживании выполнения, при «привязке» к задачам, людям, тестам, кодам.



Формальная модель требований для верификации, модельно-ориентированного тестирования и т.д.

Известны три способа разработки требований к ПО:

1. управляемая пользователем разработка;
2. контролируемая пользователем разработка;
3. независимая от пользователя разработка.

В управляемой пользователем разработке требования к ПО определяются пользователем. Такая разработка выполняется в тех случаях, когда пользователь заключает договор на разработку, и требования к ПО являются частью этого договора. Роль разработчика в формулировании этих требований сводится к выяснению их с соответствующей оценкой рассматриваемого документа. Такая разработка может приводить к созданию нескольких редакций этого документа.

В контролируемой пользователем разработке требования к ПО формулируются разработчиком при участии представителя пользователя. Роль пользователя сводится к информированию разработчика о своих потребностях и контролю за тем, чтобы формулируемые требования отражали его потребности. Разработанные требования утверждаются представителем пользователя. С точки зрения обеспечения надежности программ этот способ разработки требований является наиболее предпочтительным.

В независимой от пользователя разработке требования к ПО определяются без участия пользователя. Это происходит в том случае, когда разработчик решает создать программный продукт широкого применения.

Согласно статистике, доля ошибок в постановке требований и в определении задач системы превышает долю ошибок, допускаемых во время кодирования системы. Это объясняется субъективным характером процесса формулирования требований и отсутствием способов их формализации.

2. Виды технических спецификаций программ

Спецификация требований – законченное описание поведения программы, которую требуется разработать или, как его еще называют, внешнее описание программного обеспечения, состоящее из двух самостоятельных частей:

1) Описание поведения ПО определяет функции, которые должно выполнять приложение, и потому его называют **функциональной спецификацией ПО**. Функциональная спецификация определяет допустимые фрагменты программ, реализующих декларированные функции.

2) Требования к качеству ПО должны быть сформулированы так, чтобы разработчику были ясны цели, которые он должен стремиться достигнуть при разработке программы. Эту часть внешнего описания называют **спецификацией качества ПО.**

Обычно разработка спецификации качества предшествует разработке функциональной спецификации ПО, так как некоторые требования к качеству приложения могут предопределять включение в функциональную спецификацию специальных функций (например, защиты от несанкционированного доступа).

Спецификация качества программного средства

Под качеством программного обеспечения принято понимать совокупность характеристик, относящуюся к его способности удовлетворять установленным потребностям. Общепринятой моделью, лежащей в основе оценки качества ПО, является модель, регламентированная в стандарте ISO/IEC 9126-1:2001 «Информационная

технология. Оценка программного продукта. Характеристики качества и руководства по их применению». В соответствии с данным стандартом, модель качества ПО представляет собой иерархическую структуру, состоящую из трех уровней:

1. Характеристики качества (цели) - то, что мы хотим видеть в ПО.
2. Атрибуты качества - свойства ПО, показывающие приближение к цели.
3. Метрики - количественные характеристики степени наличия атрибутов.

Характеристики качества программного продукта:

Функциональность − способность ПО выполнять набор функций, удовлетворяющих заданным потребностям пользователей.

Надежность - это способность ПО безотказно выполнять определённые функции при заданных условиях в течение заданного периода времени с достаточно большой вероятностью. Надёжное приложение не исключает наличия в нём ошибок, важно, чтобы эти ошибки при практическом применении в заданных условиях проявлялись достаточно редко.

Легкость применения − характеристики продукта, которые позволяют минимизировать усилия пользователя по подготовке исходных данных, применению ПО и оценке полученных результатов.

Эффективность − это отношение уровня услуг, предоставляемых ПО пользователю при заданных условиях, к объему используемых ресурсов.

Сопровождаемость − характеристики ПО, которые позволяют минимизировать усилия по внесению изменений для устранения в нем ошибок и по его модификации в соответствии с изменяющимися потребностями пользователей.

Мобильность − это способность ПС быть перенесенным из одной среды (окружения) в другую, в частности, с одного компьютера на другой.

Функциональность и надежность являются обязательными характеристиками качества.

Разработка спецификации качества сводится к построению модели качества ПО. В этой модели должен быть перечень всех свойств, которыми требуется обеспечить продукт, и которые в совокупности образуют приемлемое качество программного продукта.

Каждое из этих свойств должно быть конкретизировано с точки зрения:

- оценки его наличия в ПО;
- степени преобладания в ПО.

Для конкретизации качества программ по каждой из характеристик используются **примитивы качества ПО**, регламентированные в стандарте ISO/IEC 9126.

Определения используемых примитивов качества ПО

Примитивы качества	**Свойство примитива**
Завершенность	Степень обладания необходимыми частями для выполнения своих функций
Точность	Приемлемость величины погрешности в выдаваемых программами ПО результатах
Автономность	Способность выполнять предписанные функции без помощи других компонент ПО
Независимость от устройств	Способность продукта работать на разнообразном аппаратном обеспечении (различных типах компьютеров)
Устойчивость	Способность корректного функционирования при задании неправильных входных данных
Защищенность	Способность противостоять преднамеренным или нечаянным

	деструктивным действиям пользователя
П-документированность	Наличие, полнота, понятность, доступность документации, необходимой для применения программ
Информативность	Наличие информации, необходимой для понимания назначения ПО, существующих ограничений, входных данных и результатов работы компонент, а также текущего состояния программ
Коммуникабельность	Облегчение задания входных данных, а также обеспечение выдачи сведений в понятной форме
Временная Эффективность	Способность выполнять возложенные функции за определенный отрезок времени
Эффективность по ресурсам	Способность выполнять функции при определенных ограничениях на используемые ресурсы (память)
Эффективность по устройствам	Экономичность использования устройств машины для решения поставленной задачи
С-документированность	Наличие документации, отражающей требования к ПО, и результаты различных этапов его разработки
Понятность	Степень доступности назначения, допущений и ограничений, входных данных и результатов работы программ ПО, тексты этих программ
Структурированность	Свойство, характеризующее программы ПО с точки зрения организации взаимосвязанных их частей в единое целое
Удобочитаемость	Легкость восприятия текста программ ПО (отступы, фрагментация, форматирование)
Расширяемость	Способность приложения к использованию большего объема памяти или расширению функциональных возможностей отдельных компонент
Легкость изменения	Простота внесения изменений и доработок на всех этапах и стадиях жизненного цикла ПО
Модульность	Организация программ приложения из таких дискретных компонент, что изменение одной из них оказывает минимальное воздействие на другие компоненты

Зависимость характеристик качества от примитивов качества ПС

Характеристики качества	**Примитивы качества**
Функциональность	Завершенность
Надежность	Завершенность, точность, автономность, устойчивость, защищенность
Легкость применения	П - документированность, информативность (применительно к документации по применению), коммуникабельность, устойчивость, защищенность.
Эффективность	Временная эффективность, эффективность по памяти, эффективность по устройствам
Мобильность	Независимость от устройств, автономность, структурированность, модульность
Сопровождаемость (изучаемость)	Минимизация усилий по изучению программ и документации ПО: С-документированность, информативность (применительно к С-документации),

	понятность, структурированность, удобочитаемость
Сопровождаемость (модифицируемость)	Упрощение внесений в ПО необходимых изменений и доработок: расширяемость, легкость изменения, структурированность, модульность.

Функциональная спецификация программного средства

С учетом назначения функциональной спецификации ПС она должна быть математически точной. Достаточно часто функциональная спецификация формулируется на естественном языке. Тем не менее, использование математических методов и формализованных языков при разработке функциональной спецификации весьма желательно.

Функциональная спецификация состоит из:

1. описания внешней информационной среды, к которой должны применяться программы разрабатываемого ПО;
2. описания функций ПО, определенных на множестве состояний этой информационной среды (внешние функции приложения);
3. описания исключительных ситуаций, которые могут возникнуть при выполнении программ ПС, и реакций на эти ситуации, которые должны обеспечить соответствующие программы.

В первой части определяются на концептуальном уровне:

- все используемые каналы ввода и вывода;
- все информационные объекты, к которым будет применяться разрабатываемое ПС;
- существенные связи между этими информационными объектами.

Примером описания информационной среды может быть концептуальная схема базы данных.

Во второй части вводятся обозначения всех определяемых функций; специфицируются все входные данные и результаты выполнения каждой определяемой функции, включая указание их типов и заданий всех ограничений, которым должны удовлетворять эти данные и результаты; определяется семантика каждой из этих функций.

В третьей части перечисляются все существенные случаи, когда ПС не может нормально выполнить ту или иную свою функцию:

- ошибки взаимодействия с пользователем;
- попытки применить функцию к данным, не удовлетворяющим ограничениям, указанным в ее спецификации;
- получение результата, нарушающего заданное ограничение.

Для каждого такого случая должна быть определена реакция ПО.

Таким образом, функциональная спецификация определяет:

1) что должно делать приложение;
2) какими внешними свойствами должно обладать;
3) как должно быть устроено, как обеспечить требуемые свойства;
4) определяет задачи, которые должны решить разработчики.

Суть этой спецификации должна быть понятна представителям заказчика, так как на ее основании заключается договор на разработку ПО в виде технического задания. Требования к ТЗ определяются ГОСТом 19.201 «Техническое задание. Требования к содержанию и оформлению».

Методы контроля внешнего описания ПО

Разработка внешнего описания должна завершаться проведением контроля правильности спецификаций.

Для этого используются методы:

- статический просмотр;
- смежный контроль;
- пользовательский контроль;
- имитация.

Статический просмотр – предполагает внимательное прочтение текста внешнего описания разработчиком с целью проверки его полноты и непротиворечивости, а также выявления других неточностей и ошибок.

Смежный контроль

- Контроль спецификации качества сверху - это ее проверка со стороны разработчика требований к ПС;
- Контроль функциональной спецификации - это ее проверка разработчиками требований к ПС и спецификации качества;
- Контроль внешнего описания снизу - это его изучение и проверка разработчиками архитектуры ПС и текстов программ, а также разработчиками документации по применению и разработчиками комплекта тестов.

Пользовательский контроль - этот метод контроля внешнего описания подразумевает участие пользователя (заказчика) в принятии решений при разработке внешнего описания. Если разработка требований к ПС велась под управлением пользователя, то пользовательский контроль внешнего описания, означает его смежный контроль сверху.

Имитация представляет собой своеобразный динамический контроль внешнего описания, его функциональной спецификации. Для этого необходимо подготовить тесты, и на основании функциональной спецификации осуществить имитацию работы разрабатываемого ПС.

Литература:

1. Кудлаев А.А. Лекции по технологии программирования: Основные процессы жизненного цикла программных средств. Учебное пособие. М., МИИГАиК, 2011, 71 стр. (стр. 22-27) / http://miigaik.ru/vtiaoai/tutorials/8.pdf

2. Дорофеев М. Требования к программному обеспечению и их анализ: онлайн-презентация (стр. 23-30) / https://ppt-online.org/199986