МДК. 02.01 Технология разработки программного обеспечения

Тема 2.1.1 Основные понятия и стандарти­зация требований к программному обеспе­чению

Лекция 2:

Свойства требований

**Аннотация:**В практике разработки программных систем накопились определенные представления о том, какими свойствами должны обладать требования к программной системе. В этой лекции мы рассмотрим подробнее данные свойства

Ф. Брукс в своем, теперь уже ставшим классическим, эссе1, следующим образом охарактеризовал роль требований в разработке программного обеспечения.

Строжайшее и единственное правило построения систем программного обеспечения (ПО) - решить точно, что же строить. Никакая другая часть концептуальной работы не является такой трудной, как выяснение деталей технических требований, в том числе и взаимодействие с людьми, с механизмами и с иными системами ПО. Никакая другая часть работы так не портит результат, если она выполнена плохо. Ошибки никакого другого этапа работы не исправляются так трудно.

Наука извлечения и формализации качественных (иногда говорят "хороших", "правильных") требований носит во многом эмпирический характер. Однако, в практике разработки программных систем накопились определенные представления о том, какими свойствами должны обладать требования к программной системе. Это:

* полнота,
* ясность,
* корректность,
* согласованность,
* *верифицируемость*,
* необходимость,
* полезность при эксплуатации,
* осуществимость,
* модифицируемость,
* трассируемость,
* упорядоченность по важности и стабильности,
* наличие количественной метрики.

Большинство из этих свойств раскрыто в первом разделе стандарта *IEEE*  и широко обсуждается в работах . Рассмотрим указанные выше свойства подробнее.

#### Полнота.

Как известно из теории искусственного интеллекта, неполнота - одно из фундаментальных свойств человеческого знания. При создании программных систем нам приходится иметь дело с характеристиками еще несуществующей системы. Идея о том, что необходимо сформулировать все требования полностью, т.е. исчерпывающим образом, до начала проектирования, а тем более - реализации системы, изжила себя вместе с так называемым каскадным подходом, который поддерживал последовательную модель реализации системы. Спиральный  подход, на котором базируется большинство современных методологий, предусматривает поэтапное выделение и детализацию требований на всем протяжении цикла разработки системы.

Тем не менее, требование полноты предъявляется к требованиям, формулируемым к системе. Надо понимать, что данное требование - это скорее тенденция, цель, к которой нужно постараться максимально приблизиться на как можно более ранних стадиях проекта.

Требование полноты можно рассматривать в двух аспектах: полнота отдельного требования и полнота системы требований.

Полнота отдельного требования - свойство, означающее, что текст требования не требует дополнительной детализации, то есть в нем предусмотрены все необходимые нюансы, особенности и детали данного требования.

Полнота системы требований - свойство, означающее, что совокупность артефактов, описывающих требования, исчерпывающим образом описывает все то, что требуется от разрабатываемой системы.

#### Ясность (недвусмысленность, определенность, однозначность спецификаций).

Каждый из совладельцев4 разрабатываемой системы обладает своим личным опытом восприятия событий внешнего мира. Слово, произнесенное вслух, вызывает индивидуальные ассоциации в семантическом пространстве каждого отдельного воспринимающего субъекта. То, что является ясным, допустим, для кардиохирурга, совсем необязательно будет таковым для специалиста в области программной инженерии.

Соответственно, требование обладает свойством ясности, если оно сходным образом воспринимается всеми совладельцами системы. На практике *ясность* требований достигается в том числе и в процессе консультаций, в ходе которых происходит "выравнивание *тезаурусов*" совладельцев системы. Хорошим подспорьем в этом служит согласованный сторонами *глоссарий* ключевых понятий *предметной области*.

К.Вигерс  дает следующий совет по повышению ясности документов: "Пишите документацию просто, кратко и точно, применяя лексику, понятную пользователям".

Еще одной стороной понятия "*ясность* требования" является его прослеживаемость (см. также понятие трассируемости ниже по тексту). Требование, которое сформулировано ясно, может быть прослежено, начиная от того документа, где оно сформулировано впервые, вплоть до рабочих спецификаций.

#### Корректность и согласованность (непротиворечивость).

Корректность - одно из важнейших свойств требований. К. Вигерс в  вводит понятие корректности требования через точность описания функциональности. В этом смысле корректность в определенной степени конкурирует с полнотой. Но есть и различие: если свойство полноты носит скорее качественный характер: абсолютная полнота представляет недостижимый идеал, к которому можно приближаться, то свойство корректности носит оценочный характер и задает дихотомию: каждое из требований либо корректно, либо нет. Кроме того, можно рассуждать о взаимной корректности требований или согласованности (непротиворечивости): если два требования вступают в конфликт, значит - как минимум одно из них некорректно. В иерархии требований (см. материалы "Понятие требования. Классификации требований" ) можно выделить вертикальную и горизонтальную согласованность. Иными словами, требования не должны противоречить, соответственно, требованиям своего уровня иерархии и требованиям "родительского" уровня. Так, требования пользователей не должны противоречить бизнес-требованиям, а функциональные требования - требованиям пользователя.

#### Верифицируемость (пригодность к проверке).

Признаки (свойства) требований, рассматриваемые в настоящей лекции, нельзя считать независимыми. В математической статистике такие признаки называются коррелируемыми. Так, свойство верифицируемости существенно связано со свойствами ясности и полноты: если требование изложено на языке, понятном и одинаково воспринимаемом участниками процесса создания информационной системы, причем оно является полным, т.е. ни одна из важных для реализации деталей не упущена - значит, это требование можно проверить. При этом в ходе проверки у сторон (принимающей и сдающей работу) не должно возникнуть неразрешимых противоречий в оценках. Методы верификации требований будут рассмотрены в "Проверка требований" . Так как хорошо сформулированные требования составляют основу успешного создания системы - роль верифицируемости трудно переоценить. Требования к системе представляют основу контракта между Заказчиком и Исполнителем и если данные требования нельзя проверить - значит и контракт не имеет никакого смысла, следовательно, успех или неудача проекта будут зависеть только от эмоциональных оценок сторон и их способности договориться, а это - слишком шаткая основа для осуществления работ.

#### Необходимость и полезность при эксплуатации.

Одни из самых субъективных и трудно проверяемых свойств требований.

Возвращаясь к иерархии требований в "Понятие требования. Классификации требований" , наиболее бесспорными требованиями следует считать бизнес-требования. Данные требования формулируют первые лица, представляющие Заказчика, и вряд ли кто-нибудь лучше них сможет сказать, каким условиям должна соответствовать создаваемая информационная система, чтобы соответствовать *бизнес-целям* предприятия. Тем не менее, если у представителя Исполнителя возникают сомнения в необходимости того или иного бизнес-требования, вызванные интуитивными соображениями, либо опытом внедрения информационных систем на аналогичных предприятиях, он должен проявить инициативу и собрать совместное совещание сторон. Аргументы в пользу отсутствия необходимости требования несомненно будут восприняты, особенно если они будут мотивированы в бизнес-терминологии Заказчика и подтверждены выкладками, прогнозирующими соотношение затрат на выполнение требования и ожидаемой от него эффективности.

Необходимость требований пользователя может вытекать из соответствующих бизнес-требований. Кроме того, требования пользователя могут мотивироваться эргономичностью продукта и особенностями функционирования его отдела (подразделения), недостаточно полно раскрытыми на предыдущем уровне иерархии требований.

Большинство функциональных требований вытекают из требований первых двух уровней. Другие функциональные требования могут лежать вне сферы компетенции Заказчика (который, вообще говоря, не обязан быть экспертом в области IT) и их должен сформулировать Исполнитель. Так, например, информационная система в процессе ее использования может начать снижать свою производительность из-за больших объемов накапливаемых данных. Поэтому целесообразно заложить функции архивирования информации, переключения *учетных периодов* и т.п., необходимость которых следует не из особенностей бизнеса предприятия внедрения, а из общих принципов построения информационных систем.

Более слабой, чем "необходимость" формулировкой обладает свойство "полезность при эксплуатации". Разграничение между данными свойствами можно провести следующим образом. Необходимыми следует считать свойства, без выполнения которых невозможно, либо затруднено выполнение автоматизированных бизнес-*функций пользователей*; полезными при эксплуатации следует считать любые свойства, повышающие эргономические качества продукта.

#### Осуществимость (выполнимость).

Является в некоторой степени конкурирующим по введенным выше двум свойствам.

В принципе никто не мешает сформулировать требование, выполнимость которого ограничивается сегодняшним уровнем развития техники и технологии, хотя многое из того, что было невыполнимо десять лет назад, вполне выполнимо сегодня. Можно сформулировать требование, выполнимость которого ограничена научными представлениями о строении Вселенной, например - требование мгновенной передачи информации с земной станции на Марс, хотя и фундаментальные представления иногда меняются, пусть и не так быстро, как развитие IT-технологий.

Возвращаясь с небес на землю, отринем те требования, которые можно признать абсурдными и остановимся на тех, которые выполнимы принципиально. С точки зрения науки *управления требованиями*, далеко не все из них являются осуществимыми.

Выполнимость требования на практике определяется разумным балансом между ценностью (степенью необходимости и полезности) и потребными ресурсами. Так, если стоимость контракта на разработку информационной системы составляет $10000, а затраты на выполнение нового требования, возникшее в момент, когда проект выполнен наполовину, оценивается в $4000, является ли оно невыполнимым? Скорее всего, да, если Исполнитель докажет Заказчику новизну требования (требование не входило в согласованные спецификации) и сложность его исполнения. Но, если требование является критически важным, необходимым, но выпало из поля зрения при подписании контракта Заказчик готов выделить дополнительно финансирование, а Исполнитель - трудовые ресурсы - значит, требование выполнимо. Таким образом, требование осуществимости в ряде случаев также следует считать субъективным, а критерии его оценки лежат в области договоренностей между Заказчиком и Исполнителем.

Отличной иллюстрацией балансировки между ценностью и выполнимостью требований является так называемый треугольник компромиссов.



**Рис. 3.1.**

В качестве пояснения к рисунку 3.1 приведем цитату из "белых страниц", размещенных Microsoft в открытом доступе .

Хорошо известна взаимозависимость между ресурсами проекта (людскими и финансовыми), его календарным графиком (временем) и реализуемыми возможностями (рамками). Эти три переменные образуют треугольник, показанный на рис. 3.1. После достижения равновесия в этом треугольнике изменение на любой из его сторон для поддержания баланса требует модификаций на другой (двух других) сторонах и/или на изначально измененной стороне.

Необходимо обеспечить возможность переработки требований, если понадобится, и поддерживать историю изменений для каждого положения. Для этого все они должны быть уникально помечены и обозначены, чтобы вы могли ссылаться на них однозначно. Каждое требование должно быть записано в спецификации только единожды. Иначе вы легко получите несогласованность, изменив только одно положение из двух одинаковых. Лучше используйте ссылки на первоначальные утверждения, а не дублируйте положения. Модификация спецификации станет гораздо легче, если вы составите содержание документа и указатель. Сохранение спецификации в базе данных *коммерческого инструмента* *управления требованиями* сделает их пригодными для повторного использования.

#### Трассируемость

Трассируемость требования определяется возможностью отследить связь между ним и другими артефактами информационной системы (документами, моделями, текстами программ и пр.). Отдельная трасса представляет собой направленное бинарное отношение, заданное на множестве артефактов ИС, где первый элемент отношения представляет соответствующее требование, а второй - артефакт, зависимый от данного требования. На практике трассировки анализируются при посредстве графовых, либо табличных моделей.

Процесс трассировки позволяет, с одной стороны, выявить уже на стадии проектирования системы проектные артефакты, к которым не ведет связь ни от одного из артефактов, описывающих требования, с другой - артефакты, описывающие требования, не связанные с проектными артефактами. В первом из случаев целесообразно убедиться в том, что проектный артефакт действительно имеет право на существование, а не является избыточным. Во втором случае необходимо проанализировать полезность выявленных требований: либо эти требования несут недостаточную полезную нагрузку и могут быть игнорированы, либо имеют место ошибки проектирования: пропущены соответствующие артефакты.

Другая цель трассировки - повысить управляемость проектом: при изменении отдельно взятого требования становится понятно - какие из проектных, рабочих и других артефактов подлежат изменению.

#### Упорядоченность по важности и стабильности

Приоритет требования представляет собой количественную оценку степени значимости (важности) требования. Приоритеты требований обычно назначает представитель Заказчика. Разработчик, отталкиваясь от приоритетности требований, управляет процессом реализации информационной системы.

Стабильность требования характеризует прогнозную оценку неизменности требований во времени.

#### Наличие количественной метрики

Количественные метрики играют важную роль в верификации и аттестации информационных систем. В первую очередь это относится к нефункциональным требованиям, которые, как правило, должны иметь под собой количественную основу (запрос должен отрабатываться не более, чем \_\_\_ секунд; средняя наработка на отказ должна составлять не менее, чем \_\_\_ часов). Функциональные требования также могут расширяться количественными мерами при помощи так называемых аспектов применимости.

#### Каких требований не должно быть

Согласно , спецификация требований не должна содержать деталей проектирования или реализации (кроме известных ограничений). Иными словами, требования должны отвечать на вопрос: "что должна делать система", абстрагируясь от вопроса "как она это должна делать". Стремление принимать детальные проектные решения на этапе анализа требований - одна из типичных "ловушек", типичных для неопытных команд разработчиков. Вариантов реализации всегда больше, чем один, а для принятия взвешенного решения нужна максимально более полная информация. Поэтому этапы работы с требованиями, проектирования и реализации планируются поочередно, хотя и могут быть частично запараллелены в рамках итерационного подхода к созданию программных систем