МДК. 02.01 Технология разработки программного обеспечения

Тема 2.1.1 Основные понятия и стандарти­зация требований к программному обеспе­чению

Лекция 1:

Понятие требования. Классификации требований

**Аннотация:**Существует значительное количество различных методов классификации требований, наиболее существенные из которых будут рассмотрены в лекции

### Определение понятия требования

Л.Новиков в русской редакции нотации *RUP*  приводит следующее *определение*: "Требование - это условие или возможность, которой должна соответствовать система".

В *IEEE* *Standard* *Glossary* of *Software Engineering* *Terminology* (1990)  данное понятие трактуется шире. Требование - это:

1. условия или возможности, необходимые пользователю для решения проблем или достижения целей;
2. условия или возможности, которыми должна обладать система или системные компоненты, чтобы выполнить контракт или удовлетворять стандартам, спецификациям или другим формальным документам;
3. документированное представление условий или возможностей для пунктов 1 и 2.

***IEEE*** (Institute of Electrical and Electronics Engineers) - институт инженеров по электротехнике и электронике, [http://www.ieee.org](http://www.ieee.org/) – международная некоммерческая ассоциация специалистов в области техники, делит с МЭК (<http://www.iec.ch/>) мировое лидерство в области разработки стандартов по радиоэлектронике и электротехнике. IEEE объединяет более 400 тыс. членов из 170 стран, в том числе более 100 тыс. студентов.

***IEC*** (International Electrotechnical Commission) - международная электротехническая комиссия (МЭК), [http://www.iec.ch](http://www.iec.ch/). МЭК – некоммерческая организация, наряду с IEEE ([http://www.ieee.org](http://www.ieee.org/))- признанный мировой лидер в области создания международных стандартов в сфере электрики, электронники и смежных технологий, в том числе - в области информационных технологий. Под эгидой организации сотрудничают более 10 000 специалистов. Некоторые из разработанных стандартов созданы совместно с ISO.

Введем еще одно *определение*. Требования - это исходные данные, на основании которых проектируются и создаются *автоматизированные информационные системы*. Первичные данные поступают из различных источников, характеризуются противоречивостью, неполнотой, нечеткостью, *изменчивостью. Требования* нужны в частности для того, чтобы Разработчик мог определить и согласовать с Заказчиком временные и финансовые перспективы проекта автоматизации. Поэтому значительная часть требований должна быть собрана и обработана на ранних этапах создания АИС. Однако собрать на ранних стадиях все данные, необходимые для реализации АИС, удается только в исключительных случаях. На практике процесс сбора, анализа и обработки растянут во времени на протяжении всего жизненного *цикла* АИС, зачастую нетривиален и содержит множество подводных камней; подробнее о процессе - в лекциях 4 - 8.

### Классификация требований

Существует значительное количество различных методов классификации требований, наиболее существенные из которых будут рассмотрены в лекции.

#### Требования к продукту и процессу

Требования к продукту. В своей основе требования - это то, что формулирует заказчик. Цель, которую он преследует - получить хороший конечный продукт: функциональный и удобный в использовании. Поэтому требования к продукту являются основополагающим классом требований. Более подробно требования к продукту детализируются в следующих ниже классификациях.

Требования к проекту. Вопросы формулирования требований к проекту, т.е. к тому, как Разработчик будет выполнять работы по созданию целевой системы, казалось бы, не лежат в компетенции Заказчика. Без регламентации процесса Заказчиком легко можно было бы обойтись, если бы все проекты всегда выполнялись точно и в срок. Однако, к сожалению, мировая статистика результатов программных проектов говорит об обратном. Заказчик, вступая в договорные отношения с Разработчиком, несет различные риски, главными из которых является риск получить продукт с опозданием, либо ненадлежащего качества. Основные мероприятия по контролю и снижению риска - регламентация процесса создания программного обеспечения и его аудит.

Насколько подробно Заказчику следует регламентировать требования к проекту - вопрос риторический. Ответ на него зависит от множества факторов, таких, как ценность конечного продукта для Заказчика, *степень доверия* Заказчика к Разработчику, сумма подписанного контракта, увязка срока сдачи продукта в эксплуатацию с бизнес-планами Заказчика и т.д. Однако, со всей определенностью можно сказать следующее:

1. регламентация процесса Заказчиком позволяет снизить его риски;
2. мероприятия Заказчика по регламентации процесса приводят к дополнительным накладным расходам. Требуется найти разумный компромисс между степенью контроля рисков и величиной расходов.

В качестве требований к проекту может быть внесен регламент отчетов Разработчика, совместных семинаров по оценке промежуточных результатов, определены характеристики компетенций участников рабочей группы, исполняющих проект, их количество, указана методология управления проектом. Ниже приведен пример формулировки требования к оффшорному проекту (Заказчик и Разработчик физически находятся в разных государствах) - в этой ситуации Заказчику требуется жесткий контроль над Разработчиком.

1. Разработчик представляет Заказчику согласованный план работ c детализацией (WorkBreakdownStructure - *WBS*) с точностью до конкретных исполнителей.
2. Разработчик осуществляет ежедневные сборки, *регрессионное тестирование* компонентов разрабатываемого продукта и тестирование продукта в целом.
3. Все управленческие и проектные артефакты, исходные коды и тестовые примеры размещаются в режиме online в интегрированной среде разработки Rational ClearCase с возможностью для Заказчика осуществления online-мониторинга на базе web-технологий.

***WBS*** Work Breakdown Structure – иерархическая структура работ. Согласно *PMBOK*, WBS – это ориентированная на результаты (предметы поставки) иерархическая декомпозиция работ, выполняемых командой проекта для достижения целей проекта и получения необходимых результатов. С ее помощью структурируется и определяется все содержание проекта.

#### Уровни требований

Внедрение ИС на предприятии всегда преследует конкретные *бизнес-цели* - такие, как, например, повышение прозрачности бизнеса, сокращение сроков обработки информации, экономия накладных расходов и т.д. Современные информационные системы - это крупные программные системы, содержащие в себе множество модулей, функциональных, интерфейсных элементов, отчетов и т.д. Как охватить единым взором такие разнородные вещи, как цели, преследуемые топ-менеджером предприятия Заказчика, описание интерфейса пользователя и характеристики модуля, осуществляющего расчет себестоимости изделия?

К счастью, человечество уже давно изобрело приемы борьбы со сложностью, широко применяемые в моделировании сложных объектов - абстракцию и декомпозицию. Применительно к дисциплине анализа требований к программным системам эти принципы работают следующим образом. Требования разделяются по уровням. Уровни требований связаны, с одной стороны, с уровнем абстракции системы, с другой - с уровнем управления на предприятии.

Обычно выделяют три уровня требований.

* На верхнем уровне представлены так называемые бизнес-требования (*business requirements*). Примеры бизнес-требования: система должна сократить срок оборачиваемости обрабатываемых на предприятии заказов в три раза. Бизнес-требования обычно формулируются топ-менеджерами, либо *акционерами* предприятия.
* Следующий уровень - уровень требований пользователей (*user requirements*). Пример требования пользователя: система должна представлять диалоговые средства для ввода исчерпывающей информации о заказе, последующей фиксации информации в базе данных и маршрутизации информации о заказе к сотруднику, отвечающему за его планирование и исполнение. Требования пользователей часто бывают плохо структурированными, дублирующимися, противоречивыми. Поэтому для создания системы важен третий уровень, в котором осуществляется формализация требований.
* Третий уровень - функциональный (*functional requirements*). Пример функциональных требований (или просто функций) по работе с электронным заказом: заказ может быть создан, отредактирован, удален и перемещен с участка на участок.

Существуют объективные противоречия между требованиями различных уровней. Так, очевидным бизнес-требованием является требование о полноте информации, собираемой на рабочих местах пользователей в единую базу данных. Чем полнее информация - тем глубже база для анализа деятельности и принятия решений. С другой стороны, конкретному пользователю системы вполне может быть достаточно использования только той части информации, которая влияет на выполнение его основных функций.

Важные правила внедрения и использования АИС на предприятии - "Одна точка сбора", "Данные собираются там, где они появляются". Использование этих правил позволяет избежать затрат на необоснованное дублирование информации и, что важнее - потерь от ошибок учета, неизбежно возникающих при дублировании точек ввода.

Внедрение АИС на предприятии приводит к необходимости оснащения всех точек ввода информации автоматизированными рабочими местами (АРМ), обучению персонала и, зачастую, оптимизации и повышению уровня формализации рабочих процессов, выполняемых персоналом. Поэтому внедрение АИС - непростой процесс, часто требующий "перекройки человеческого материала" и встречающий сопротивление со стороны пользователей, которые не готовы, либо не хотят работать по-новому.

#### Системные требования и требования к программному обеспечению

Существуют различные трактовки понятия "Системные требования" (*system requirements*).

К. Вигерс формулирует данный термин, как "высокоуровневые требования к продукту, которые содержат многие подсистемы, то есть системе" . При этом под системой понимается программная, программно-аппаратная, либо *человеко-машинная система*. Данная система является сложной, структурированной системой и системные требования являются подмножеством функциональных требований к продукту. В данное подмножество целесообразно относить наиболее важные, существенные требования, которые относятся в целом к системе и не содержат избыточной детализации.

INCOSE (International Council on *Systems Engineering*) дает более детальное определение системы: "комбинация взаимодействующих элементов, созданная для достижения определенных целей; может включать аппаратные средства, программное обеспечение, встроенное ПО, другие средства, людей, информацию, техники (подходы), службы и другие поддерживающие элементы". Таким образом, происходит разделение между системными требованиями, как обобщающему понятию и требованиями к программному обеспечению, как выделенному подмножеству системных требований, направленных исключительно на программные компоненты системы. Этот же подход прослеживается в стандарте ГОСТ Р ИСО/МЭК 12207/99: работы, связанные с системой в целом и с программным обеспечением выделяются в отдельные группы в целях удобства оперирования.

***INCOSE*** (International Council on Systems Engineering) - Международный совет по системной инженерии, [www.incose.org](http://www.incose.org) — некоммерческая организация, ставящая своей целью развитие системной инженерии и профессиональный рост системных инженеров. В настоящее время насчитывает более 8000 членов. Под эгидой INCOSE разработан ряд международных стандартов в области системной инженерии.

В практике компьютерной инженерии бытует другой, более узкий контекст использования данного понятия: под системными требованиями в узком смысле понимаются требования, выдвигаемые прикладной программной системой (в частности - информационной) к среде своего функционирования (системной, аппаратной). Пример таких требований - тактовая частота процессора, объем памяти, требования к выбору операционной системы.

#### Функциональные, нефункциональные требования и характеристики продукта

Функциональные требования регламентируют функционирование или поведение системы (behavioral requirements). Функциональные требования отвечают на вопрос "что должна делать система" в тех или иных ситуациях. Функциональные требования определяют основной "фронт работ" Разработчика, и устанавливают цели, задачи и сервисы, предоставляемые системой Заказчику.

Функциональные требования записываются, как правило, при посредстве предписывающих правил: "система должна позволять кладовщику формировать приходные и *расходные накладные*". Другим способом являются так называемые варианты использования (users cases) - популярный и весьма продуктивный способ *представления требований*.

***Use case*** - вариант использования, прецедент. Данный термин был введен в обиход программной инженерии шведским учёным Айваром Якобсоном (Ivar Hjalmar Jacobson) и по сей день является одной из наиболее позитивных абстракций в области создания требований к программному обеспечению. Согласно нотации UML 2.4.1 (<http://www.omg.org/spec/UML/2.4.1/>), прецеденты являются средством для определения требуемых использований системы. Как правило, они применяются для извлечения требований к системе, то есть, того, что система предполагает делать. Основными понятиями, связанными с вариантами использования являются акторы, прецеденты, и объект. Объектом является рассматриваемая система, в которой применяются прецеденты. Пользователи и любые другие системы, которые могут взаимодействовать с объектом, представлены в качестве акторов. Акторы всегда моделируют сущности, находящиеся за пределами системы. Требуемое поведение объекта задается одним или несколькими вариантами использования, которые определяются в соответствии с потребностями акторов. Строго говоря, термин "вариант использования" относится к типу прецедента. Экземпляр прецедента относится к проявлению поведения, соответствующего типу прецедента. Такие случаи, как правило, описывается через спецификацию взаимодействий.

Это - основной, определяющий вид требований, который будет рассматриваться на протяжении всего лекционного курса.

Нефункциональные требования, соответственно, регламентируют внутренние и внешние условия или атрибуты функционирования системы. К. Вигерс  выделяет следующие основные группы нефункциональных требований:

* Внешние интерфейсы (*External Interfaces*),
* Атрибуты качества (*Quality Attributes*),
* Ограничения (Constraints).

Среди внешних интерфейсов в большинстве современных АИС наиболее важным является интерфейс пользователя (User Interface, UI). Кроме того, выделяются интерфейсы с внешними устройствами (аппаратные интерфейсы), программные интерфейсы и интерфейсы передачи информации (коммуникационные интерфейсы).

Основные атрибуты качества:

* Применимость,
* Надежность,
* Производительность,
* Эксплуатационная пригодность,

достаточно хорошо раскрыты в модели FURPS (см. ниже).

Ограничения  - формулировки условий, модифицирующих требования или наборы требований, сужая выбор возможных решений по их реализации. Выбор платформы реализации и/или развертывания (протоколы, серверы приложений, баз данных, ...), которые, в свою очередь, могут относиться, например, к внешним интерфейсам (конец цитаты).

Характеристики продукта. К.Вигерс  формулирует характеристику, "фичу" (feature), как набор логически связанных функциональных требований, которые обеспечивают возможности пользователя и удовлетворяют *бизнес-цели*.

Существует и более общий взгляд на данное понятие: "features могут быть как относящимися к функциональным, так и к нефункциональным требованиям и могут изменяться от версии к версии продукта".

С.Орлик в  отмечает, что "с точки зрения инженерии требований, features являются самостоятельным артефактом, который может быть соотнесен как с функциональными требованиями, так и с нефункциональными".

Роль характеристик проявляется в первую очередь в области маркетинга: не всякий потенциальный потребитель продукта станет читать его функциональные описания, а набор ключевых характеристик, характеризующих конкурентные преимущества, можно сделать лаконичным и уместить на одной странице рекламной листовки, либо напечатать на компакт-диске.

#### Классификация RUP

В спецификациях Rational Unified Process при классификации требований используется модель FURPS+ со ссылкой на стандарт *IEEE* Std 610.12.1990 .

Акроним FURPS обозначает следующие категории требований:

* Functionality (Функциональность)
* Usability (Применимость)
* Reliability (Надежность)
* Performance (Производительность)
* Supportability (эксплуатационная пригодность).

Символ "+" расширяет FURPS-модель, добавляя к ней:

* ограничения проекта,
* требования выполнения,
* требования к интерфейсу,
* физические требования,

часть из которых уже была рассмотрена выше.

Кроме того, в спецификациях RUP выделяются такие категории требований, как

* требования, указывающие на необходимость согласованности с некоторыми юридическими и нормативными актами;
* требования к лицензированию,
* требования к документированию.

***FURPS*** (Functionality Usability Reliability Performance Supportability: функциональность, удобство использования, надежность, производительность, удобство сопровождения)— классификация требований к программным системам, разработанная в Hewlett-Packard. Согласно классификации, все требования подразделяются на 5 категорий, непосредственно следующих из составляющих наименования классификации. В настоящее время используется, как составная часть более общей классификации FURPS+.

***FURPS+*** (Functionality Usability Reliability Performance Supportability +: функциональность, удобство использования, надежность, производительность, удобство сопровождения, дополнительные требования) — расширенная версия классификации требований FURPS. Дополнительно включает в себя ограничения, разделенные на следующие группы требований:

* ограничения проектирования (design);
* ограничения разработки (implementation);
* ограничения на интерфейсы (interface);
* физические ограничения (physical).

Используется в методологии *RUP*.

Подробно описана в работе Роберта Грейди.

### Методологии и стандарты, регламентирующие работу с требованиями

Среди основополагающих нормативных документов в области работы с требованиями можно выделить следующие.

1. Разработки *IEEE*:

* IEEE 1362 "Concept of Operations Document".
* IEEE 1233 "Guide for Developing *System Requirements* Specifications".
* IEEE Standard 830-1998, "IEEE Recommended Practice for Software Requirements Specifications"
* IEEE Standard *Glossary* of *Software Engineering* *Terminology*/IEEE Std 610.12-1990
* IEEE Guide to the *Software Engineering* Body of Knowledge (1) - *SWEBOK*®, 2004.

2. Отечественные ГОСТ:

* ГОСТ 34.601-90. Информационная технология. Автоматизированные системы. Стадии создания.
* ГОСТ 34.602-89. Информационная технология. Техническое задание на создание автоматизированной системы
* ГОСТ 19.201-78. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению.