**Лекция 10. Программная инженерия**

**Разработка требований к программным системам**

**10.1 Определение программных требований (ПТ) и процесса их разработки (requirements engineering)**

**ПТ** — это совокупность свойств ПО, которые надлежащим образом представлены для решения конкретных практических задач.

**Причины, из-за которых процесс выработки ПТ является сложным и ответственным**:

* если программный продукт является инновационным (впервые (для исполнителя) создаваемый для новой предметной области), то трудно чётко сформулировать те действия, которые должна выполнять система;
* на ПО налагаются конкретные ограничения по **аппаратным** и **архитектурным** особенностям, **среде функционирования** и **масштабированию**.

**Проблемы ИТ-проектов, вызванные плохой организацией процесса выработки ПТ**

Чаще всего проблемами, с которыми встретились не достигшие своих целей проекты программных продуктов, являются: недостаток информации от пользователя или заказчика о функциях проекта, неполные, некорректные требования, а также **многочисленные изменения требований и** **спецификаций**.

Это происходит потому, что зачастую разработчики и заказчики считают, что «даже если мы не очень точно знаем, чего хотим достичь, лучше быстрее приступить к реализации проекта, так как мы и так выбились из графика и нам некогда размышлять. Мы можем уточнить требования позднее».

Подобный подход приводит к хаотическим, неупорядоченным действиям при разработке ПС, когда никто не знает, что на самом деле хочет заказчик и пользователь и как в **действительности** функционирует созданная на **данный момент** система и/или программный продукт.

**Формализация** и **управлени**е требованиями — это систематический метод **выявления**, **организации** и **документирования** требований к системе и/или ПС а также **процесс**, в ходе которого вырабатывается и обеспечивается соглашение между заказчиком и выполняющими проект специалистами, в условиях **меняющихся** (!) требований к системе (рис.10.1).

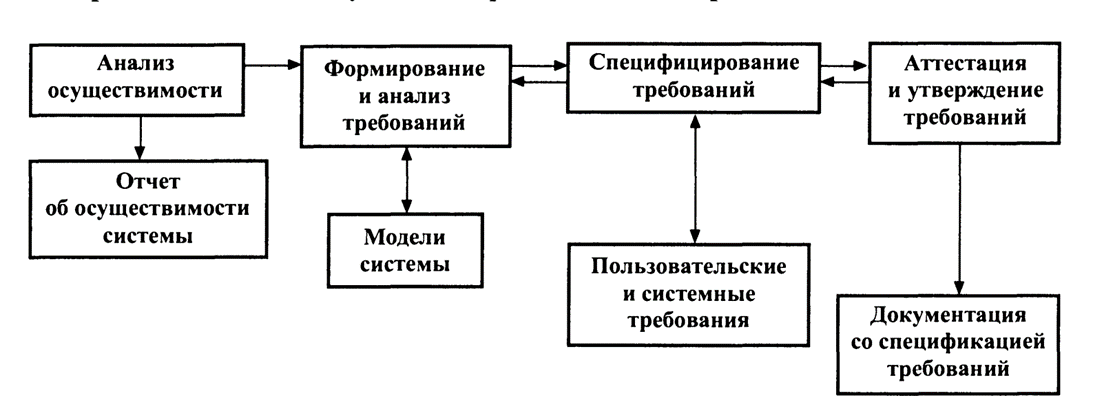


Рис. 10.1. Общая схема взаимосвязи этапов процесса разработки требований

Только **систематическая работа** с требованиями позволяет надёжно выполнить следующие задачи, необходимые для решения задач выполнения требований заказчика:

* моделирование задач предметной области заказчика («реального мира»)
* разработку (формулирование) необходимых рабочих приёмочных тестов валидации ПП по критериям, заданным реальными практическими требованиями заказчика.

**Каждый** (**?**) член команды в той или иной степени должен привлекаться к управлению и формализации требований к проекту. Команде необходимо выработать профессиональные приемы для понимания **потребностей** пользователей, **управления масштабом** ПС, **структурой** и **построением** **системы**, удовлетворяющей эти потребности.

**10.2. Определение групп (типов, категорий) и уровней основой для заключения контракта между разработчиком и заказчиком (требований**

Согласно К. Вигерсу[[1]](#footnote-1):

1. Системные;
2. Программные;
3. Функциональные;
4. Нефункциональные.

**Уровни требований (классификация по уровням)**:

1. **Пользовательские** — для обозначения **высокоуровневых обобщённых требований**. Описание на естественном языке с возможным использованием диаграмм функций, выполняемых системой. Определяют набор пользовательских задач, которые должна решать программа, а также способы (сценарии) их решения в системе.

Они могут выражаться в виде **фраз утверждений, в виде сценариев использования (англ. use case), пользовательских историй (англ. user stories), сценариев взаимодействия (scenario).**

2. **Системные** — для **детализированного** описания выполняемых системой функций. Служит основой для заключения контракта между исполнителем и заказчиком (покупателем) ПО.

3. **Проектная системная спецификация** — обобщённое описание структуры программной системы, которое будет основой для более детализированного проектирования системы и её последующей реализацией. «Мост» между этапом разработки требований и этапом проектирования системы.

4.**Бизнес- требования** - определяют назначение ПО, описываются в документе о видении (vision) и границах проекта (scope).

**Ещё одна используемая классификация**:

1. Функциональные;

2.Нефункциональные;

3.Требования предметной области — также разделяются на **функциональные** и **нефункциональные**.

**Виды требований по характеру**

***Функциональный характер — требования к поведению системы***

1.Бизнес-требования

2.Пользовательские требования

3.Функциональные требования

**Нефункциональный характер** — требования к **характеру** поведения системы

1.Бизнес-правила — определяют ограничения, проистекающие из предметной области и свойств автоматизируемого объекта (предприятия)

2.**Системные требования и ограничения** — определения элементарных операций, которые должна иметь система, а также различных условий, которым она может удовлетворять. К системным требованиям и ограничениям относятся:

3.Ограничения на программные интерфейсы, в том числе к внешним системам

4.Требования к атрибутам качества

5.Требования к применяемому оборудованию и ПО

6.Требования к документированию

7.Требования к дизайну и юзабилити

8.Требования к безопасности и надёжности

9.Требования к показателям назначения (производительность, устойчивость к сбоям и т. п.)

10.Требования к эксплуатации и персоналу

11.Прочие требования и ограничения (внешние воздействия, мобильность, автономность и т. п.).

**Графическое представление требований** – связи с необходимой информацией

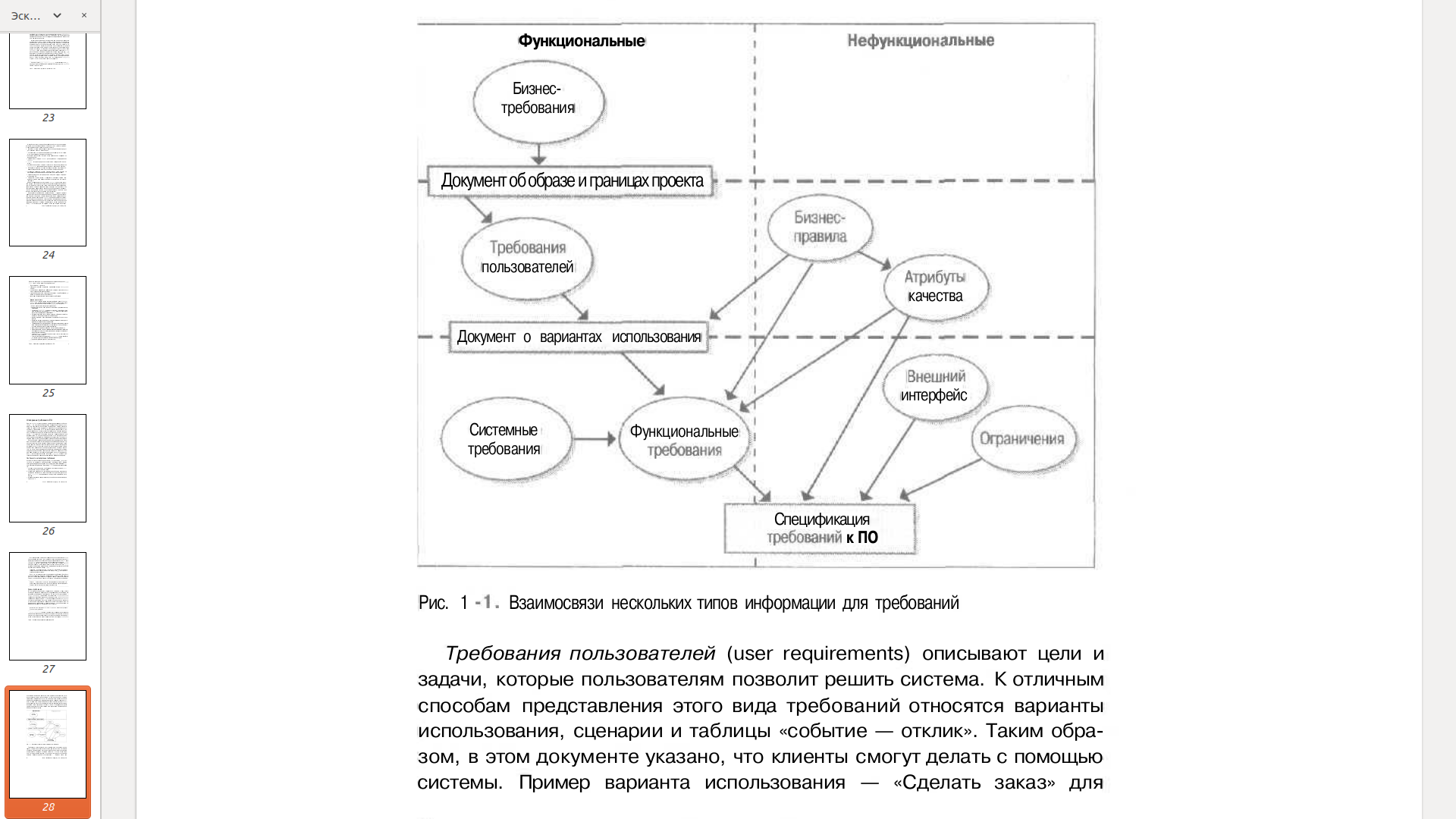
****

Рис. 10.2. Группа информации, необходимых для выработки требований



Рис. 10.2. Группы требований, предъявляемых к ПП

**Чёткой границы в классификациях не существует**.

**10.3 Описание требований**

***10.3.1 Функциональные требования*** - требования к **поведению** системы (?)

Описывают поведение системы и сервисы, которые она выполняет.

**Сервисы** – это функции.

Если функциональные требования оформлены как **пользовательские**, они описывают поведение системы в **обобщённом** виде (как правило).

Если **функциональные требования оформлены как системные**, **они описывают поведение системы максимально подробно**. То есть входные и выходные данные, исключения и. т. д. Именно такие требования могут описываться разными способами.

[https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B5%D0%B1%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F\_%D0%BA\_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%BC%D1%83\_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8E](https://ru.wikipedia.org/wiki/Требования_к_программному_обеспечению) !!!

***10.3.2 Нефункциональные требования***- пожалуй, более сложное понятие!

Определяют требования к **характеру поведения** системы. Не связаны непосредственно с функциями предметной области, выполняемые системой.

* Надёжность;
* Время отклика на запрос, время реакции;
* Ограничения, накладываемы на систему: пропускная способность устройств ввода-вывода, форматы данных, используемые в **системном** интерфейсе и т. д. – смотри выше.

**Нефункциональные требования могут относиться как к отдельным средствам системы, так и к системе в целом**. Они более значимы и критичны (особенно последние), чем **отдельные** функциональные требования. Ошибка в нефункциональных требованиях может сделать систему неработоспособной, а ошибка в функциональных – просто снизить качество системы.

Существуют нефункциональные требования, относящиеся не к системе.

Например, такие требования:

* к технологическому процессу её создания,
* **содержать перечень стандартов**, **используемых** в процессе разработки,
* чтобы проектирование системы выполнялось с использованием определённого case-средства,
* о предварительном описании процесса проектирования.

Как и функциональные, нефункциональные требования отображают (соответствуют) потребностям пользователя, но с системной, архитектурной стороны. При этом в отличие от функциональных, они учитывают:

* организационные возможности компании-разработчика,
* возможности взаимодействия разрабатываемой системы с другими программными и вычислительными системами,
* правила техники безопасности,
* законодательство в области информационной безопасности, о защите интеллектуальной собственности и т. п.

На рис. 10.3 показана квалификация нефункциональных требований.

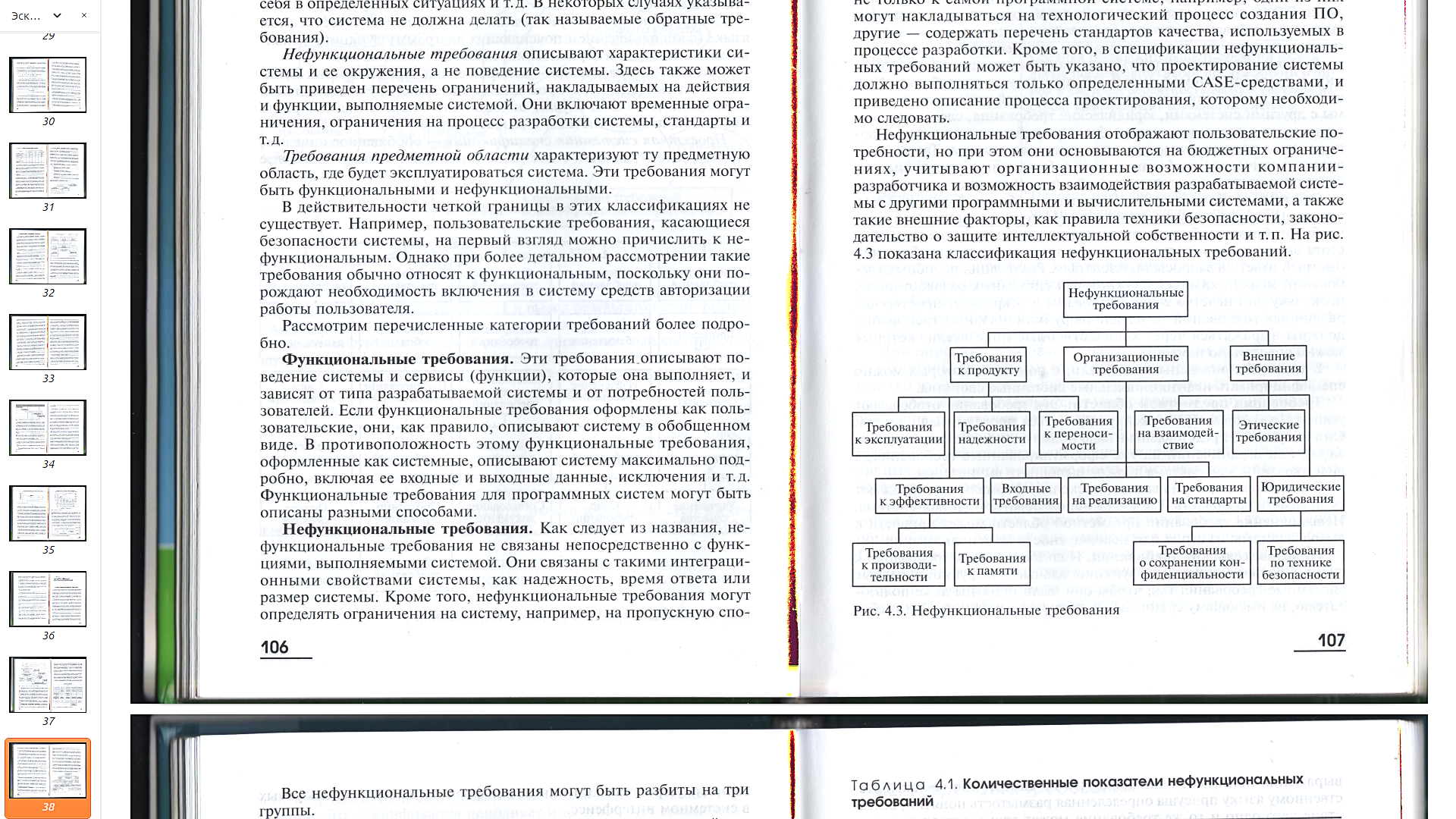


Рис. 10.3. Нефункциональные требования

**Требования к продукту** — требования к его эксплуатационным свойствам.

К ним относятся:

* производительность,
* объём необходимой памяти,
* частота возможных сбоев в системе (надёжность),
* переносимость системы на разные компьютерные платформы,
* удобство эксплуатации.

**Организационные требования** отображают политику и организационные процедуры как заказчика, так и разработчика ПО. Включают стандарты разработки ПО, требования к языку программирования, сроки изготовления ПО и требования к сопутствующей документации.

**Внешние требования** — для учёта факторов, внешних по отношению к системе и процессу разработки. Это:

* условия взаимодействия с другими системами, уже работающими у заказчика;
* юридические требования, гарантирующие законность разработки и использования заказанной системы;
* **этические требования -?**

Нефункциональные требования достаточно трудно проверяются. Достаточно часто они отображают общие цели заказчика:

* простота эксплуатации;
* возможность восстановления после сбоя;
* быстрый ответ на запросы пользователя и т.д.

Желательно (для системных разработчиков) выражать их через количественные показатели. Примеры такого подхода приведены в таблице 10.1.

Таблица 10.1.

Примеры количественных показателей **нефункциональных требований**

|  |  |
| --- | --- |
| Показатель | Единица измерения |
| Скорость | Число выполненных транзакций в секунду;  Время реакции на действия пользователя;  Время обновления экрана и т.д. |
| Размер | Килобайты, число модулей занимаемой памяти |
| Простота эксплуатации | Время обучения персонала;  Объём справочного материала |
| Надёжность | Средняя продолжительность времени между двумя последовательными проявлениями ошибок в системе;  Вероятность системы выхода из строя;  Коэффициент готовности системы |
| Устойчивость к сбоям | Время восстановления системы после сбоя;  Процент событий, приводящих к сбоям;  Вероятность порчи данных при сбое |
| Переносимость | Процент машинно-зависимых операторов;  Число машинно-зависимых подсистем |

**Требования предметной области**.

Должны отображать условия, в которых будет эксплуатироваться система. Это новые функциональные требования или ограничения на сформулированные функциональные требования. Примеры (**?**).

В качестве примера рассмотрим требования к библиотечной системе.

1. Стандартный пользовательский интерфейс, предоставляющий доступ ко всем библиотечным базам данных, должен основываться на стандарте Z39.50.
2. Для обеспечения авторских прав некоторые документы должны быть удалены из системы сразу после получения. Для этого, в зависимости от желания пользователя, эти документы могут быть распечатаны или на локальном системном сервере, или на сетевом принтере.

Первое требование является ограничением на **системное функциональное требование**. Оно указывает, что пользовательский интерфейс к базам данных должен быть реализован согласно соответствующему библиотечному стандарту. Второе требование является внешним и направлено на выполнение закона об авторских правах, применяемого к библиотечным материалам. Из этого требования вытекает, что система должна иметь средство "удалить\_на\_печать", применяемое автоматически для некоторых типов библиотечных документов.

В следующем примере сформулированы требования предметной области, указывающие, как должны выполняться вычисления. **Они взяты из спецификации системы автоматического торможения поезда**. Эта система должна автоматически останавливать поезд на красный сигнал семафора. Данное требование указывает способ вычисления скорости поезда при торможении. Здесь использована терминология, применяемая при расчетах скоростей поезда. Чтобы разобраться в ней, необходимы соответствующие знания о системах управления поездами и их характеристиках.

**Пример: Торможение поезда вычисляется по формуле**

Dпоезд = Dуправление + Dградиент

Приведенный пример показывает основную проблему, связанную с требованиями предметной области. **Требования этого типа используют язык и обозначения, присущие данной предметной области, что затрудняет их понимание разработчиками ПО**. **Вследствие этого требования предметной области не всегда выполняются так, как подразумевается заказчиками программной системы.**

**Пользовательские требования**.

Пользовательские требования к системе должны описывать **функциональные и нефункциональные системные требования** так, чтобы они были понятны даже пользователю, не имеющему специальных технических знаний. Эти требования **должны** определять только **внешнее поведение** системы, избегая по возможности определения структурных характеристик системы. Пользовательские требования должны быть написаны естественным языком с использованием простых таблиц, а также наглядных и понятных диаграмм.

Вместе с тем при описании требований на естественном языке могут возникнуть различные проблемы.

1. ***Отсутствие четкости изложения***. Иногда нелегко изложить какую-либо мысль естественным языком четко и недвусмысленно, не сделав при этом текст многословным и трудночитаемым.
2. ***Смешение требований***. В пользовательских требованиях отсутствует четкое разделение на функциональные и нефункциональные требования, на системные цели и проектную информацию.
3. ***Объединение требований*.** Несколько различных требований к системе могут описываться как единое пользовательское требование.

Чтобы свести к минимуму неясности при написании пользовательских требований, в [**Соммервилл Инженерия программного обеспечени**я] приведены следующие рекомендации.

1. Разработайте стандартную форму для записи пользовательских требований и неукоснительно ее придерживайтесь. Стандартная форма записи уменьшает неясности в формулировке требований и позволяет легко их проверить. Я рекомендую включать в форму записи требования не только саму его формулировку, но его обоснование и ссылку на более детализированную спецификацию требований.
2. Делайте различие между **обязательными** и **описательными** требованиями, как показано в примере 4. Здесь обязательным требованием является наличие средства добавления новых структурных элементов, описательным — описание последовательности действий пользователя. Описательное требование не является абсолютно необходимым для реализации данного пользовательского требования и при необходимости может быть изменено.
3. Используйте разные начертания шрифта (полужирное и курсив) для выделения ключевых частей требования.

Избегайте по возможности компьютерного жаргона. Это не исключает использования технических терминов той предметной области, для которой разрабатывается программное обеспечение.

**Системные требования**.

Это более детализированное описание пользовательских требований. **Они обычно служат основой для заключения контракта на разработку программной системы** и поэтому должны представлять максимально **полную спецификацию системы в целом**. Системные требования также используются в качестве отправной точки на этапе проектирования системы.

Спецификация системных требований может строиться на основе различных системных моделей, таких, как объектная модель или модель потоков данных. Различные модели, используемые при разработке спецификации системных требований, рассматриваются в лекции 6.

**В принципе системные требования определяют, что должна делать система, не показывая при этом механизма ее реализации**. Но, с другой стороны, для полного описания системы требуется детализированная информация о ней, которая по возможности должна включать всю информацию о системной архитектуре. На то существует ряд причин.

1. Первоначальная архитектура системы помогает структурировать спецификацию требований. Системные требования должны описывать подсистемы, из которых состоит разрабатываемая система.
2. В большинстве случаев разрабатываемая система должна взаимодействовать с уже существующими системами. Это накладывает определенные ограничения на архитектуру новой системы.
3. В качестве внешнего системного требования может выступать условие использования для разрабатываемой системы специальной архитектуры.

<https://vc.ru/books/86775-shest-superknizhek-dlya-nachinayushchih-specialistov-v-sfere-analiza-trebovaniy-k-po>

# Заключение

**Требования программной системы** – это описание того, что система должна делать, а также ограничений, накладываемых на ее поведение и реализацию.

**Функциональные требования** – описания сервисов, предоставляемых системой, и способов выполнения вычислительных операций. Требования предметной области – это функциональные требования, которые вытекают из характеристик той **предметной области**, где будет эксплуатироваться разрабатываемая система.

**Нефункциональные требования**- это ограничения, накладываемые на систему, на процесс разработки системы, а также внешние требования. Они описывают свойства системы в целом.

**Пользовательские требования** предназначены для людей, которые будут эксплуатировать систему. Они должны писаться естественным языком с использованием таблиц и диаграмм, простых для восприятия.

**Системные требования** должны **максимально точно описывать функции, выполняемые системой.**

**Определение необходимых понятий**

**Валидация** в технике или в системе менеджмента качества — **процесс приведения доказательств** того, что требования конкретного внешнего потребителя или пользователя продукта, услуги или системы удовлетворены.

**Верифика́ция** (от лат. verum «истинный» + facere «делать») в различных сферах деятельности человека может подразумевать: **проверка, подтверждение, метод доказательств** каких-либо теоретических положений, алгоритмов, программ и процедур путём их сопоставления с опытными.

1. К.Вигерс. Разработка требований к программному обеспечению. изд-во: БХВ-ПетербургISBN: 978-5-9775-3348-5, год издания: **2019**, 736 стр.

   Аннотация: Эта книга — подробное руководство по разработке качественных требований к программному обеспечению. Здесь описаны десятки проверенных на практике приемов выявления, формулирования, разработки, проверки, утверждения и тестирования требований, которые помогут разработчикам, менеджерам и маркетологам создать эффективное ПО. Настоящее издание дополнено новыми приемами, посвященными разработке требований в проектах гибкой разработки (agile). Основная аудитория — **бизнес-аналитики и разработчики**, а также дизайнеры, программисты, тестировщики и другие члены команды, задача которых понять и удовлетворить чаяния клиентов, а также маркетологи, менеджеры по продуктам и менеджеры проекта, которые должны проникнуться «духом» и особенностями продукта, чтобы сделать его в полной мере конкурентоспособным. Книга состоит из 32 глав, 3 приложений и словаря терминов. [↑](#footnote-ref-1)