ЛЕКЦИЯ 4
ИНЖЕНЕРИЯ ТРЕБОВАНИЙ

Требования к системе. Функциональные и нефункциональные требования. Пользовательские требования. Системные требования. Документирование системных требований.

Содержание лекции

4.1. О термине «требования». Синтаксис требований.
4.2. Уровни требований к системе.
4.3 Источники требований к системе.
4.4.Требования к программному обеспечению.
4.5. Функциональные и нефункциональные требования.
4.6. Требования предметной области.
4.7. Пользовательские требования.
4.8. Системные требования.

4.1. О термине «требования»

Необходимо обратить внимание на следующие определения понятия “требование” (на основе работ Вигерса и стандарта IEEE Standard Glossary of Software Engineering Terminology, 1990):

Условие или возможность, требуемая пользователем для решения задач или достижения целей.
Условие или возможность, которые должны удовлетворяться системой/компонентом системы или которыми система/компонент системы должна обладать для обеспечения условий контракта, стандартов, спецификаций или др. регулирующими документами.
Документальная репрезентация (зафиксированное представление, определение, описание) условий или возможностей, перечисленных в предыдущих пунктах.

По справочнику Батоврина [1. Батоврин В.К. Системная и программная инженерия. Словарь-справочник: учеб. пособие для вузов. – М.: ДМК Пресс, 2010. – 280 с.]:

Требование – это условие или возможность, которым должна отвечать или которыми должна обладать система (элемент системы), для того, чтобы удовлетворять контракту, стандарту, спецификации или другому формально одобренному документу. **ISO/IEC 24765.**

Синтаксис требований
[обстоятельства] [субьект] [действие] [объект] [ограничение]
Пример:
Когда сигнал получен [обстоятельства] система [субьект]
должна установить[действие] разряд сигнала [объект]
в течение двух секунд [ограничение]
или:
[обстоятельство] [действие] [значение]
Пример:
В состоянии 1 [обстоятельство]
минимальный диапазон должен быть [действие]
не менее 8 миль [значение]

4.2. Уровни требований к системе

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.
Второй 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.

4.2. Источники требований к системе.

Требования могут выдвигаться различными сторонами. Из Википедии источники требований:
Федеральное и муниципальное отраслевое законодательство (конституция, законы, распоряжения)
Нормативное обеспечение организации (регламенты, положения, уставы, приказы)
Текущая организация деятельности объекта автоматизации
Модели деятельности (диаграммы бизнес-процессов)
Представления и ожидания потребителей и пользователей системы
Журналы использования существующих программно-аппаратных систем
Конкурирующие программные продукты

При разработке требований к системе следует принимать во внимание мнение ВСЕХ возможных пользователей. Сложные системы получают требования из многих источников (рис.4.1). Даже ОДНО неучтенное требование может привести к большим проблемам или печальному результату.

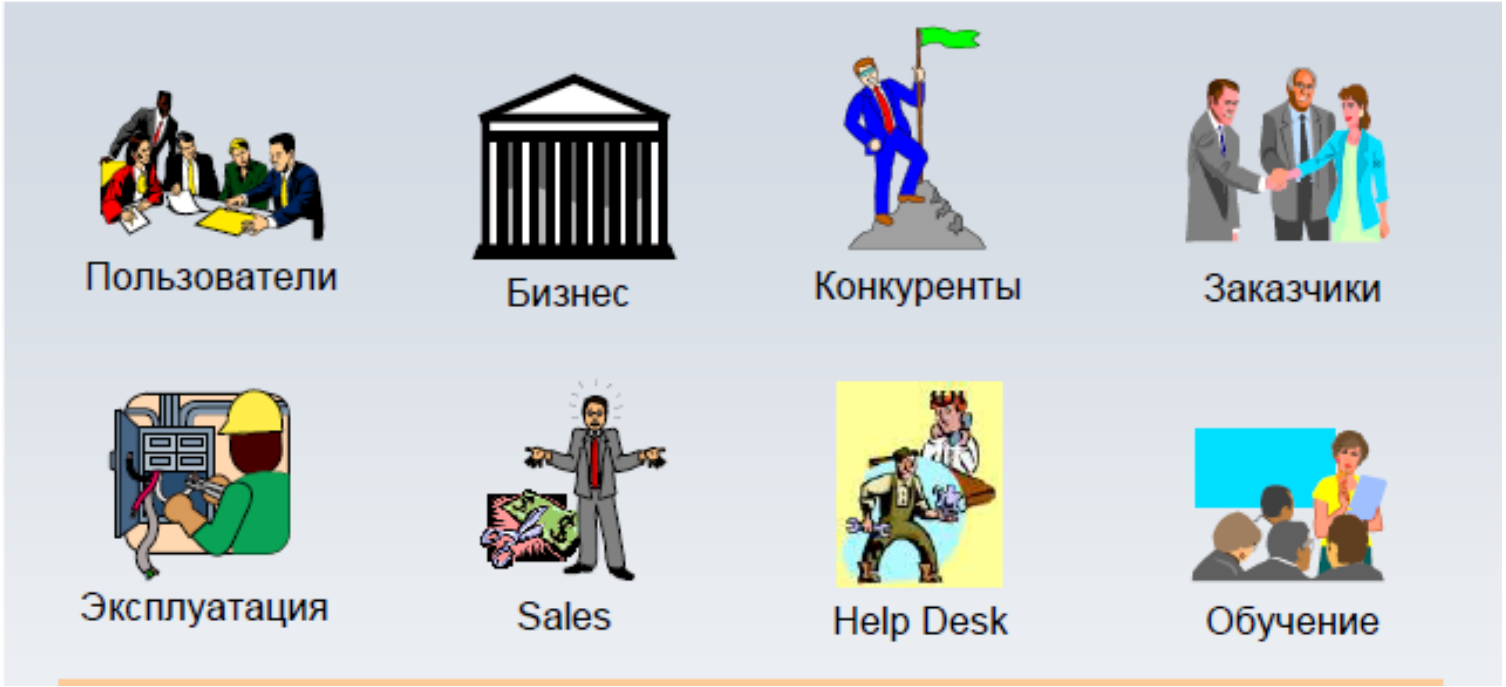


Рис. 4.1. Источники требований

sales manager - 1) коммерческий директор 2) заведующий отделом сбыта.

Help desk в дословном переводе означает стол помощи. В английском языке help desk – это устоявшееся сочетание, обозначающее техническую поддержку компьютерного парка. Таким образом служба технической поддержки – это help desk.

Технические требования разрабатываются в целях:
Конкретизации и дифференциации требований к составу, структуре и функциям оборудования на объектах, в зависимости от их статуса и назначения;
Установления требований к методам и техническим средствам, применяемых на объектах, и их техническому и метрологическому обеспечению;
Конкретизации требований к программно-техническому комплексу автоматизированных систем управления;
Определения порядка, организации и проведения метрологического контроля и надзора за состоянием и применением средств измерений и методик выполнения измерений.

Результатами анализа и разработки требований могут быть:

Техническое задание;

Технические требования;

Общие технические решения;

Задание на проектирование;

Подготовка конкурсной документации.

Совет: как создавать требования

Принимайте во внимание различные источники (внутренние, внешние, Web).

Идентифицируйте типы и группы пользователей.

Общайтесь с каждым.

Попытайтесь заставить пользователя выражать свои мысли в терминах процессов и данных, используемых ими на каждом шаге разработки

Записывайте каждое требование как полное предложение, сформулированное в утвердительное форме

Не забывайте о прошлых ошибках и старайтесь обойти их хорошими и простыми альтернативными вариантами требований

Постарайтесь выяснить природу возникновения требования.

Не стесняйтесь на некоторые требования заказчика спросить - ПОЧЕМУ

Никогда не оставляйте попыток улучшить формулировку требования.
Остановитесь только когда каждый скажет, что понял, что имеется в виду
Не жалейте времени сформулировать требование как можно более однозначно и недвусмысленно. Скорость работы многих специалистов - одна страница в час.
Потратьте и вы хотя бы столько же времени, чтобы создать хороший документ с требованиями – это окупится сторицей.

Такой подход начинается с понимания потребностей заказчика, определения функциональности изделия и обязательных запланированных проверок (аттестаций, приемочных испытаний, контроля) на самых ранних стадиях жизненного цикла создания изделия

Преимущества, которые дает управление требованиями
Информированность – ясное понимание целей и задач разработки
Прозрачность – руководство может видеть общую картину и статус проекта
Тестируемость – известно что тестировать, чтобы сдать продукт заказчику
Интеграция – все отдельные блоки и модули наконец-то работают вместе
Трассируемость – прозрачность отношений между требованиями
Управление изменениями – оценка последствий вносимого изменения
Оптимизация – разрабатываем и поставляем только то, что заказывалось
Качество – мы хорошо понимаем, как много это значит для бизнеса
Удовлетворение - заказчик и бизнес получают то, что хотели
Соответствие – демонстрация соответствия нормативным документам
Анализ - возможность оперативного принятия решений

Таким образом, системная инженерия отвечает за всю картину в целом, обеспечивая выполнение требований в течение всего жизненного цикла изделия.

4.4. ТРЕБОВАНИЯ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

Проблемы, которые приходится решать специалистам в процессе создания программного обеспечения, обычно очень сложны. Природа этих проблем не всегда ясна, особенно если разрабатываемая программная система инновационная. В частности, трудно четко описать те действия, которые должна выполнять система. Описание функциональных возможностей и ограничений, накладываемых на программную систему, называется требованиями к этой системе, а сам процесс формирования, анализа, документирования и проверки этих функциональных возможностей и ограничений — разработкой требований (requirements engineering). Термин требования (к программной системе) может трактоваться по-разному. В некоторых случаях под требованиями понимаются высокоуровневые обобщенные утверждения о функциональных возможностях и ограничениях системы. Другая крайняя ситуация— детализированное математическое формальное описание системных функций

Например, если компания хочет выиграть контракт на разработку большого программного проекта, она вынуждена, пока решение не принято, представлять требования в самом обобщенном виде, чтобы, с одной стороны, удовлетворить требования заказчика, а с другой — иметь возможность для маневра при конкуренции с другими компаниями-разработчиками. После того как контракт выигран, компания должна представить заказчику более подробное описание системы с указанием всех выполняемых ею функций. В обеих ситуациях предоставляются документы, которые называются документированными требованиями к системе.

На практике часто применяется подход, используемый в различных методологиях разработки ПО и базирующийся на определении групп требований к продукту. Такой подход обычно включает группы (типы, категории) требований, например: системные, программные, функциональные, нефункциональные (в частности, атрибуты качества) и т.п.

Некоторые проблемы, возникающие в процессе разработки требований, порождены отсутствием четкого понимания различия между этими разными уровнями требований. Чтобы различить требования разных уровней, здесь используются термины пользовательские требования (user requirements) для обозначения высокоуровневых обобщенных требований и системные требования (system requirements) для детализированного описания выполняемых системой функций. Кроме требований этих двух уровней, применяется еще более детализированное описание системы — проектная системная спецификация (software design specification), которая может служить мостом между этапом разработки требований и этапом проектирования системы. Три перечисленных вида требований можно определить следующим образом.

Пользовательские требования— описание на естественном языке (плюс поясняющие диаграммы) функций, выполняемых системой, и ограничений, накладываемых на нее.
Системные требования. — детализированное описание системных функций и ограничений, которое иногда называют функциональной спецификацией. Она служит основой для заключения контракта между покупателем системы и разработчиками ПО.
Проектная системная спецификация— обобщенное описание структуры программной системы, которое будет основой для более детализированного проектирования системы и се последующей реализации. Эта спецификация дополняет и детализирует спецификацию системных требований.
Различие между пользовательскими и системными требованиями показаны в примере, представленном примере 1. Здесь показано, как пользовательские требования могут быть преобразованы в системные.

Пример 1. Пользовательские и системные требования

Пользовательские требования

1. ПО должно предоставить средство доступа к внешним файлам, созданным в других программах.

Спецификация системных требований

Пользователь должен иметь возможность определять тип внешних файлов.

Для каждого типа внешнего файла должно иметься соответствующее средство, применимое к этому типу файлов.

Внешний файл каждого типа должен быть представлен соответствующей пиктограммой на дисплее пользователя.

Пользователю должна быть предоставлена возможность самому определять пиктограмму для каждого типа внешних файлов.

При выборе пользователем пиктограммы, представляющей внешний файл, к этому файлу должно быть применено средство, ассоциированное с внешними файлами данного типа.

Пользовательские требования пишутся для заказчика ПО и для лица, заключающего контракт на разработку программной системы, причем они могут не иметь детальных технических знаний по разрабатываемой системе (рис. 4.2). Спецификация системных требований предназначена для руководящего технического состава компании-разработчика и для менеджеров проекта. Она также необходима заказчику ПО и субподрядчикам по разработке. Эти оба документа также предназначены для конечных пользователей программной системы. Наконец, проектная системная спецификация является документом, который ориентирован на разработчиков ПО.

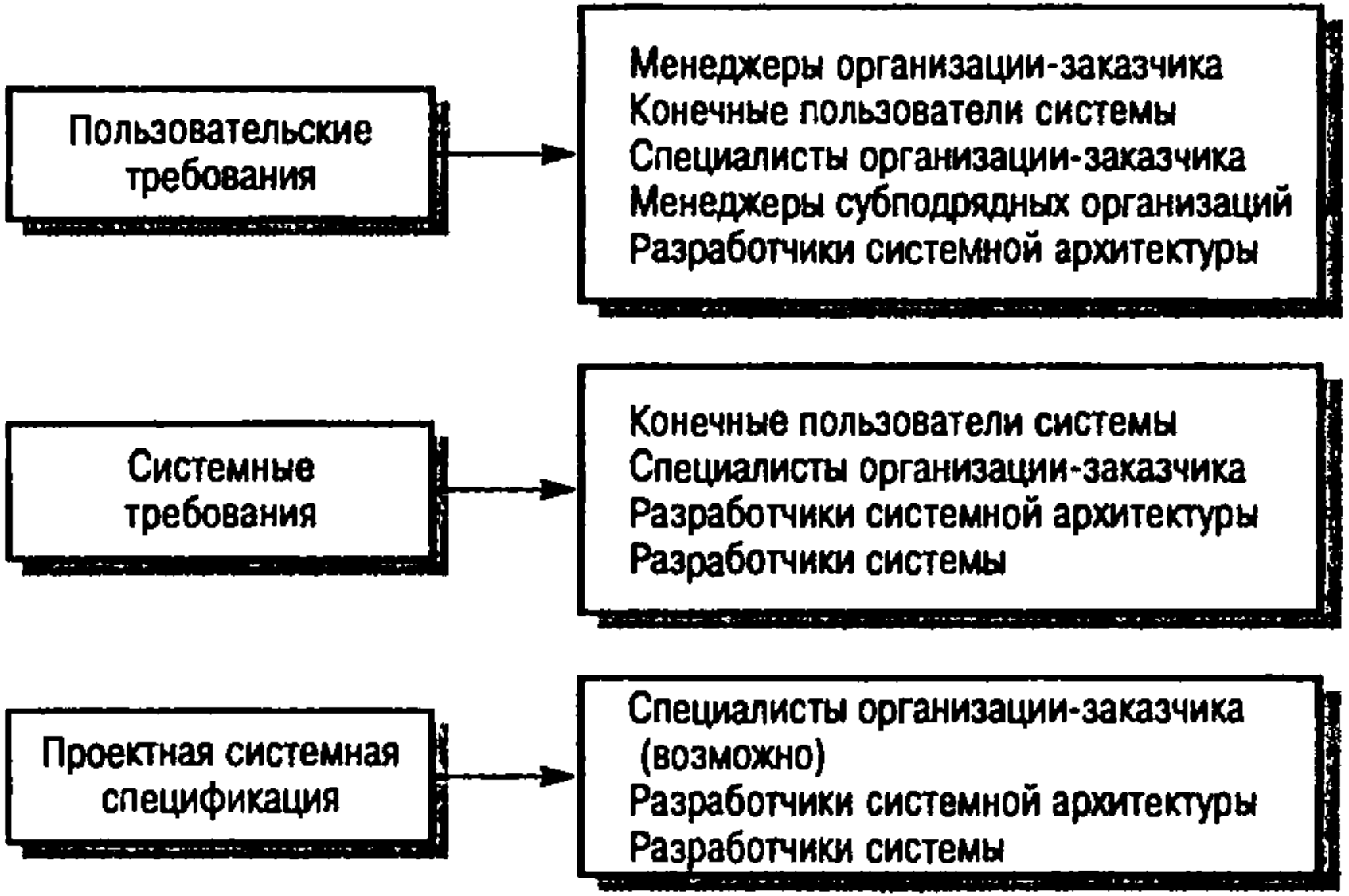


Рис.4.2. Различные типы спецификаций требований и их читатели

4.5. ФУНКЦИОНАЛЬНЫЕ И НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ (Functional and Non-functional Requirements)

Требования к программной системе часто классифицируются как функциональные, нефункциональные и требования предметной области. **Функциональные требования** задают “что” система должна делать; **нефункциональные** – с соблюдением “каких условий” (например, скорость отклика при выполнении заданной операции); часто функциональные требования представляют в виде сценариев (вариантов использования) Use Case.

Функциональные требования. Это перечень сервисов, которые должна выполнять система, причем должно быть указано, как система реагирует на те или иные входные данные, как она ведет себя в определенных ситуациях и т.д. В некоторых случаях указывается, что система не должна делать.

Нефункциональные требования. Описывают характеристики системы и ее окружения, а не поведение системы. Здесь также может быть приведен перечень ограничений, накладываемых на действия и функции, выполняемые системой. Они включают временные ограничения, ограничения на процесс разработки системы, стандарты и тд.

Требования предметной области. Характеризуют ту предметную область, где будет эксплуатироваться система. Эти требования могут быть функциональными и нефункциональными.

В действительности четкой границы между этими типами требований не существует. Например, пользовательские требования, касающиеся безопасности системы, можно отнести к нефункциональным. Однако при более детальном рассмотрении такое требование можно отнести к функциональным, поскольку оно порождает необходимость включения в систему средства авторизации пользователя. Поэтому, рассматривая далее эти виды требований, мы должны всегда помнить, что данная классификация в значительной степени искусственна.

Классический пример (см. рисунок 4.3) высокоуровневого структурирования групп требований как требований к продукту описан в работах одного из классиков дисциплины управления требованиями – Карла Вигерса.

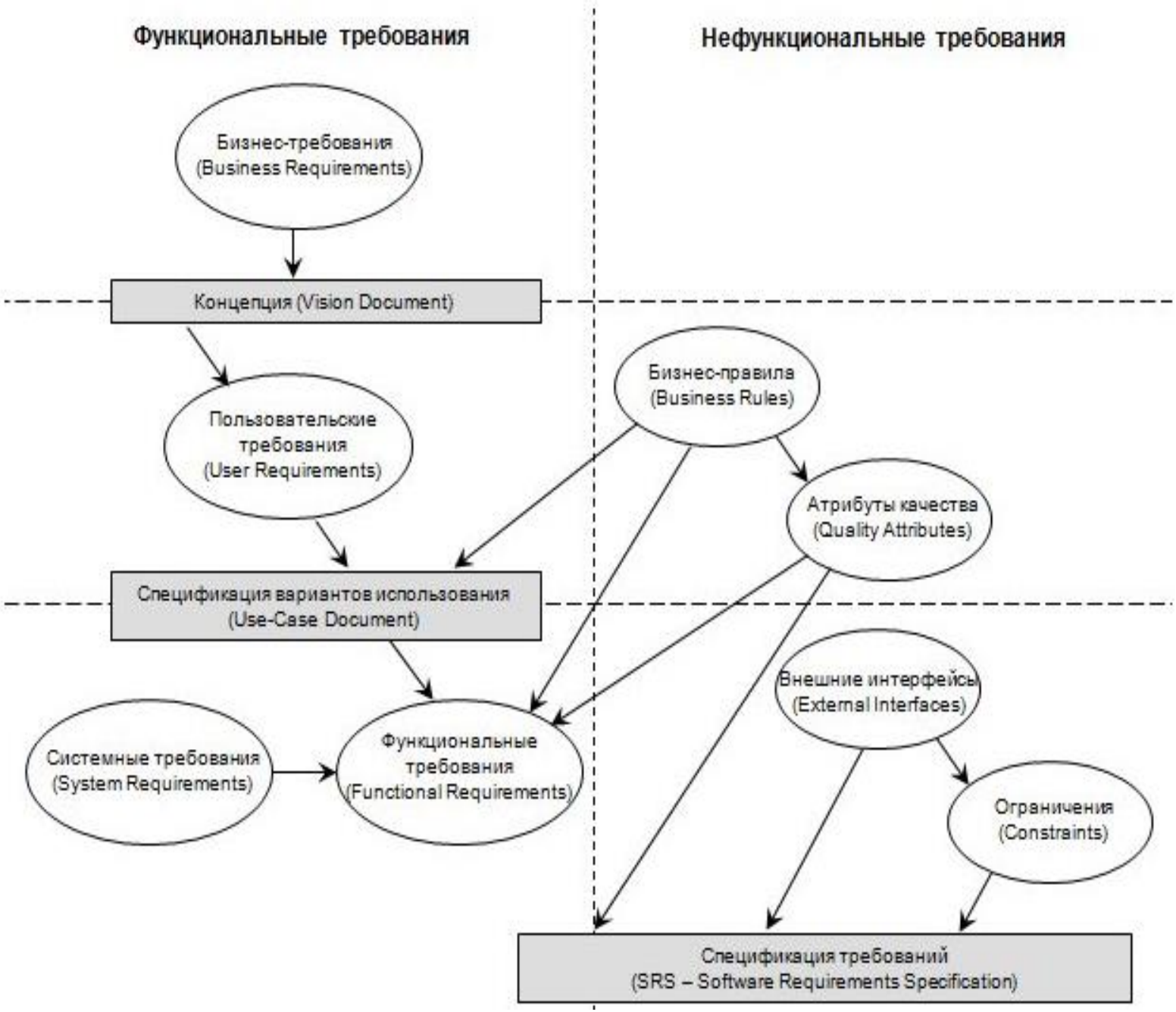


Рисунок 4.3. Уровни требований по Вигерсу

Группа функциональных требований

Бизнес-требования (Business Requirements) – определяют высокоуровневые цели организации или клиента (потребителя) – заказчика разрабатываемого программного обеспечения.

Пользовательские требования (User Requirements) – описывают цели/задачи пользователей системы, которые должны достигаться/выполняться пользователями при помощи создаваемой программной системы. Эти требования часто представляют в виде *вариантов использования (Use Cases)*.

Функциональные требования (Functional Requirements) – определяют функциональность (поведение) программной системы, которая должна быть создана разработчиками для предоставления возможности выполнения пользователями своих обязанностей в рамках бизнес-требований и в контексте пользовательских требований

Группа нефункциональных требований (Non-Functional Requirements)

Бизнес-правила (Business Rules) – включают или связаны с корпоративными регламентами, политиками, стандартами, законодательными актами, внутрикорпоративными инициативами (например, стремление достичь зрелости процессов по CMMI 4-го уровня), учетными практиками, алгоритмами вычислений и т.д. На самом деле, достаточно часто можно видеть недостаточное внимание такого рода требованиям со стороны сотрудников ИТ-департаментов и, в частности, технических специалистов, вовлеченных в проект. Business Rules Group дает понимание *бизнес-правила,* как “положения, которые определяют или ограничивают некоторые аспекты бизнеса. Они подразумевают организацию структуры бизнеса, контролируют или влияют на поведение бизнеса”. Бизнес-правила часто определяют распределение ответственности в системе, отвечая на вопрос “кто будет осуществлять конкретный вариант, сценарий использования” или диктуют появление некоторых функциональных требований. *В контексте дисциплины управления проектами* (уже вне проекта разработки программного обеспечения, но выполнения бизнес-проектов и бизнес-процессов) такие правила обладают высокой значимостью и, именно они, часто определяют ограничения бизнес-проектов, для автоматизации которых создается соответствующее программное обеспечение.

Внешние интерфейсы (External Interfaces) – часто подменяются “пользовательским интерфейсом”. На самом деле вопросы организации пользовательского интерфейса безусловно важны в данной категории требований, однако, конкретизация аспектов взаимодействия с другими системами, операционной средой (например, запись в журнал событий операционной системы), возможностями мониторинга при эксплуатации – все это не столько функциональные требования (к которым ошибочно приписывают иногда такие характеристики), сколько вопросы интерфейсов, так как функциональные требования связаны непосредственно с *функциональностью* системы, направленной на решение *бизнес-потребностей*.

Атрибуты качества (Quality Attributes) – описывают дополнительные характеристики продукта в различных “измерениях”, важных для пользователей и/или разработчиков. Атрибуты касаются вопросов портируемости, интероперабельности (прозрачности взаимодействия с другими системами), целостности, устойчивости и т.п.

Ограничения (Constraints) – формулировки условий, модифицирующих требования или наборы требований, сужая выбор возможных решений по их реализации. В частности, к ним могут относиться параметры производительности, влияющие на выбор платформы реализации и/или развертывания (протоколы, серверы приложений, баз данных, ...), которые, в свою очередь, могут относиться, например, к внешним интерфейсам.

Системные требования (System Requirements) – иногда классифицируются как составная часть группы функциональных требований (не путайте с как таковыми “функциональными требованиями”). Описывают высокоуровневые требования к программному обеспечению, содержащему несколько или много взаимосвязанных подсистем и приложений. При этом, система может быть как целиком программной, так и состоять из программной и аппаратной частей. В общем случае, частью системы может быть персонал, выполняющий определенные функции *системы*, например, авторизация выполнения определенных операций с использованием программно-аппаратных подсистем.

4.1.1. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

Эти требования описывают поведение системы и сервисы (функции), которые она выполняет, и зависят от типа разрабатываемой системы и от потребностей пользователей. Если функциональные требования оформлены как пользовательские, они, как правило, описывают системы в обобщенном виде. В противоположность этому функциональные требования, оформленные как системные, описывают систему максимально подробно, включая ее входные и выходные данные, исключения и т.д. Функциональные требования для программных систем могут быть описаны разными способами. Рассмотрим для примера функциональные требования к библиотечной системе университета, предназначенной для заказа книг и документов из других библиотек.

Пользователь должен иметь возможность проводить поиск необходимых ему книг и документов или по всему множеству доступных каталожных баз данных или по определенному их подмножеству.
Система должна предоставлять пользователю подходящее средство просмотра библиотечных документов.
Каждый заказ должен быть снабжен уникальным идентификатором (ORDERJD), который копируется в формуляр пользователя для постоянного хранения.

Эти функциональные пользовательские требования определяют свойства, которыми должна обладать система. Они взяты из документа, содержащего пользовательские требования, и показывают, что функциональные требования могут быть описаны с разным уровнем детализации (сравните первое и третье требования).

Многие проблемы, возникающие при разработке систем, связаны с неточностью и "размытостью" спецификации требований. Естественно, разработчики интерпретируют требования, допускающие двоякое толкование, так, чтобы систему было проще реализовать. Но это толкование может не совпадать с ожиданиями заказчика. Такая ситуация приводит к разработке новых требований и внесению изменений в систему. Это, в свою очередь, ведет к задержке сдачи готовой системы и ее удорожанию.

Рассмотрим второе требование к библиотечной системе из приведенного выше списка и обратим внимание на выражение "подходящее средство просмотра документов". Библиотечная система может предоставлять документы в широком спектре форматов. В требовании подразумевается, что система должна предоставить средства для просмотра документов в любом формате. Но поскольку это условие четко не выписано, разработчики в случае дефицита времени могут использовать простое средство для просмотра текстовых документов и настаивать на том, что именно такое решение следует из данного требования.

В принципе спецификация функциональных требований должна быть комплексной и непротиворечивой. Комплексность подразумевает описание (определение) всех системных сервисов. Непротиворечивость означает отсутствие несовместимых и взаимоисключающих определений сервисов. На практике для больших и сложных систем крайне трудно разработать комплексную и непротиворечивую спецификацию функциональных требований. Причина кроется частично в сложности самой разрабатываемой системы, а частично — в несогласованных опорных точках зрения на то, что должна делать система. Эта несогласованность может не проявиться на этапе первоначального формулирования требований — для ее выявления необходим более глубокий анализ спецификации. Когда несогласованность системных функций проявится на каком-либо этапе жизненного цикла программы, в системную спецификацию придется внести соответствующие изменения.

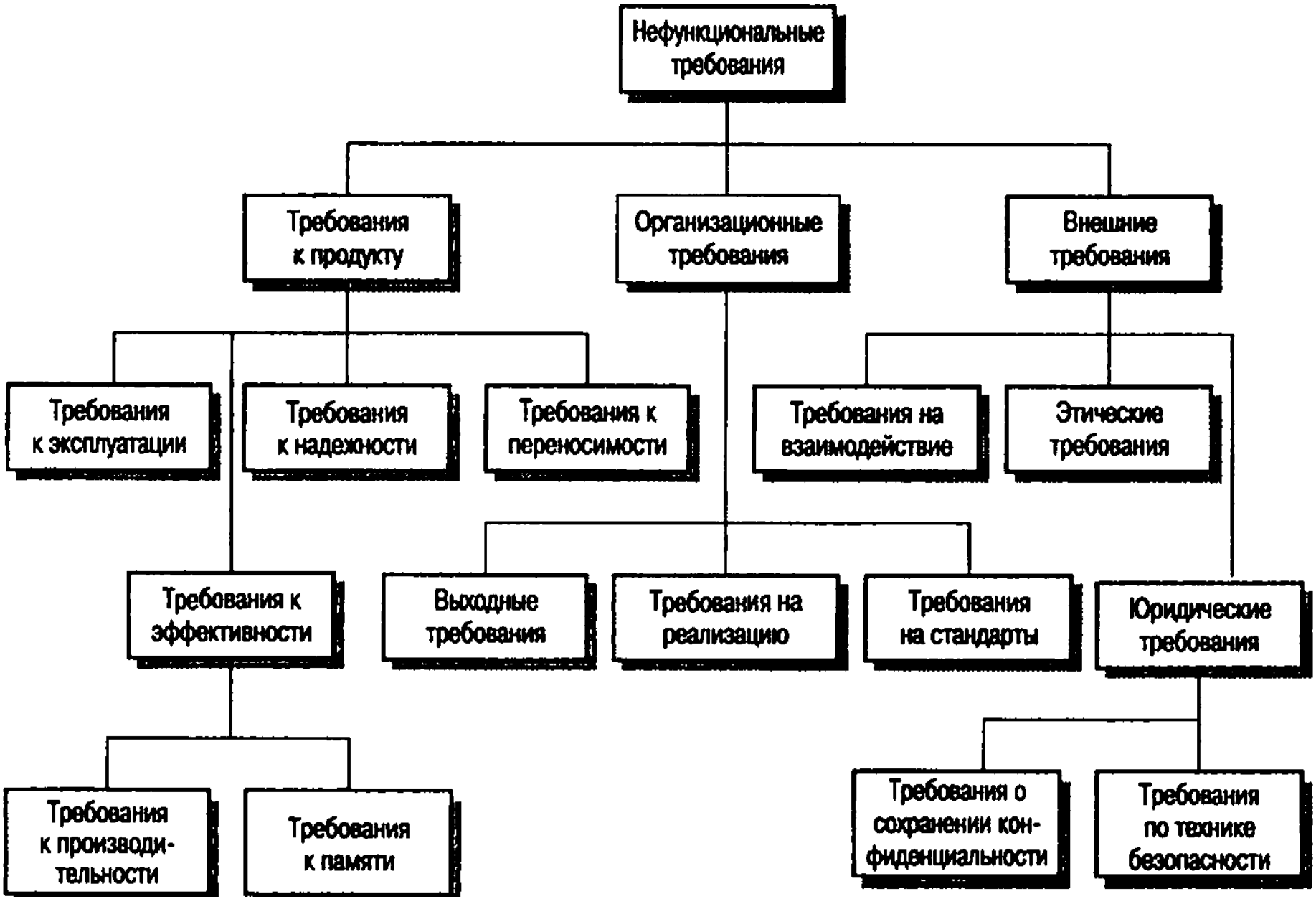
4.1.2. НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

Как следует из названия, нефункциональные требования не связаны непосредственно с функциями, выполняемыми системой. Они связаны с такими интеграционными свойствами системы, как надежность, время ответа или размер системы. Кроме того, нефункциональные требования могут определять ограничения на систему, например на пропускную способность устройств ввода-вывода, или форматы данных, используемых в системном интерфейсе.

Многие нефункциональные требования относятся к системе в целом, а не к отдельным ее средствам. Это означает, что они более значимы и критичны, чем отдельные функциональные требования. Ошибка, допущенная в функциональном требовании, может снизить качество системы, ошибка в нефункциональных требованиях может сделать систему неработоспособной. Вместе с тем нефункциональные требования могут относиться не только к самой программной системе: одни могут относиться к технологическому процессу создания ПО, другие — содержать перечень стандартов качества, накладываемых на процесс разработки. Кроме того, в спецификации нефункциональных требований может быть указано, что проектирование системы должно выполняться только определенными CASE-средствами, и приведено описание процесса проектирования, которому необходимо следовать.

Нефункциональные требования отображают пользовательские потребности; при этом они основываются на бюджетных ограничениях, учитывают организационные возможности компании-разработчика и возможность взаимодействия разрабатываемой системы с другими программными и вычислительными системами, а также такие внешние факторы, как правила техники безопасности, законодательство о защите интеллектуальной собственности и т.п. На рис. 4.4 показана классификация нефункциональных требований.

Все нефункциональные требования, показанные на рис.4.4. разбиты на три большие группы.


Нефункциональные требования
Требования к продукту
Организационные требования
Внешние требования
Требования к эксплуатации
Требования к надежности
Требования к переносимости
Требования на взаимодействие
Этические требования
Требования к эффективности
Выходные требования
Требования на реализацию
Требования на стандарты
Юридические требования
Требования к производительности
Требования к памяти
Требования о сохранении конфиденциальности
Требования по технике безопасности

Требования к продукту. Описывают эксплуатационные свойства программного продукта. Сюда относятся требования к производительности системы, объему необходимой памяти, надежности (определяет частоту возможных сбоев в системе), переносимости системы на разные компьютерные платформы и удобству эксплуатации.

Организационные требования. Отображают политику и организационные процедуры заказчика и разработчика ПО. Они включают стандарты разработки программного продукта, требования к реализации ПО (т.е. к языку программирования и методам проектирования), выходные требования, которые определяют сроки изготовления программного продукта, и сопутствующую документацию.

Внешние требования. Учитывают факторы, внешние по отношению к разрабатываемой системе и процессу ее разработки. Они включают требования, определяющие взаимодействие данной системы с другими системами, юридические требования, следование которым гарантирует, что система будет разрабатываться и функционировать в рамках существующего законодательства, а также этические требования. Последние должны гарантировать, что система будет приемлемой для пользователей или заказчика.

Основная проблема нефункциональных требований состоит в том, что их выполнение трудно проверить. Часто они пишутся для того, чтобы отобразить общие цели заказчика системы, такие, как простота эксплуатации, возможность восстановления после сбоев или быстрый ответ на запросы пользователя. Реализация подобных требований может оказаться сложной для системных разработчиков, поскольку они нечетко сформулированы и открывают простор для различных толкований. Подобную ситуацию иллюстрирует пример 2. Здесь одним из основных показателей (целей) системы указана простота эксплуатации, что в виде нефункциональных требований можно выразить различными способами. В данном случае требование сформулировано так, что его можно проверить.

Пример 2. Системные цели и проверка требований.

Системная цель.

Система должна быть простой, в эксплуатации для опытного оператора и сводить количество его ошибок к минимуму.

Проверяемое нефункциональное требование.

Опытному оператору должны быть доступны все системные функции после двух часов обучения работе с данной системой. После такого обучения число ошибок оператора не должно превышать двух за рабочий день.

В идеале нефункциональные требования должны выражаться через количественные показатели, которые можно объективно измерить. В табл. 4.1 приведены показатели, с помощью которых можно специфицировать нефункциональные системные свойства. На практике выразить нефункциональные требования с помощью количественных показателей весьма затруднительно. Часто заказчик ПО не может оформить свое видение будущей системы посредством требований, выраженных количественными показателями. Либо некоторые системные требования, например удобство сопровождения, вообще нельзя выразить через количественные показатели. Кроме того, затраты на объективное измерение количественных нефункциональных требований могут оказаться крайне высокими.

Нефункциональные требования часто вступают в конфликт с другими требованиями, предъявляемыми системе. Например, в соответствии с одним из системных требований размер системы не должен превышать 4 Мбайт, поскольку она должна полностью поместиться в постоянное запоминающее устройство ограниченной емкости. Другое требование обязывает использовать для написания системы язык программирования Ada, который часто применяется для создания критических систем реального времени. Но, допустим, откомпилированная системная программа, написанная на языке Ada, занимает более 4 Мбайт. Итак, одновременное выполнение этих требований невозможно. В этой ситуации следует отказаться от одного из требований. Можно или применить другой язык программирования, или увеличить объем памяти, выделяемый для системы.

Таблица 4.1. Количественные показатели нефункциональных требований

Показатель	Единицы измерения
Скорость	Количество выполненных транзакций в секунду; время реакции на действия пользователя; время обновления экрана
Размер	Килобайты; количество модулей памяти
Простота эксплуатации	Время обучения персонала; количество статей в справочной системе
Надежность	Средняя продолжительность времени между двумя последовательными проявлениями ошибок в системе; вероятность выхода системы из строя; коэффициент готовности системы
Устойчивость к сбоям	Время восстановления системы после сбоя; процент событий, приводящих к сбоям; вероятность порчи данных при сбоях
Переносимость	Процент машинно-зависимых операторов; количество машинно-зависимых подсистем

4.6. ТРЕБОВАНИЯ ПРЕДМЕТНОЙ ОБЛАСТИ.

Эти требования отображают условия, в которых будет эксплуатироваться программная система. Они могут быть представлены в виде новых функциональных требований, в виде ограничений на уже сформулированные функциональные требования или в виде указаний, как система должна выполнять вычисления. Эти требования очень важны, поскольку отображают ту предметную область, где будет использоваться данная система. Невыполнение требований предметной области может привести к выходу системы из строя.
В качестве примера рассмотрим требования к библиотечной системе (см. раздел 4.1.1).

Стандартный пользовательский интерфейс, предоставляющий доступ ко всем библиотечным базам данных, должен основываться на стандарте Z39.50.
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 примере 3 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 3. 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.

Пример 3. 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.

4.7. 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.

Отсутствие четкости изложения. Иногда нелегко изложить какую-либо мысль естественным языком четко и недвусмысленно, не сделав при этом текст многословным и трудночитаемым.

Смешение требований. В пользовательских требованиях отсутствует четкое разделение на функциональные и нефункциональные требования, на системные цели и проектную информацию.

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 обеспечения] приведены следующие рекомендации.

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.

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.

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.

Пример 4.
Пользовательские требования по созданию структурных элементов схемы

Добавление структурных элементов в схему
Редактор должен иметь средство, предоставляющее пользователю возможность добавлять в схему новые структурные элементы выбранного типа.
'Последовательность действий пользователя для добавления в схему нового структурного элемента
Пользователь выбирает тип добавляемого элемента.
Пользователь помещает курсор в нужную позицию на схеме и указывает, каким символом будет отображаться новый элемент.
Пользователь перемещает символ элемента в конечную позицию.

Обоснование. Такой подход к реализации функции добавления новых структурных элементов предоставляет пользователю непосредственный контроль над выбором типа элемента и его позиционированием на схеме.

4.8. СИСТЕМНЫЕ ТРЕБОВАНИЯ.

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

Спецификации системных требований часто пишутся естественным языком. Но использование естественного языка может породить определенные проблемы при написании детализированной спецификации. Применение естественного языка подразумевает, что те, кто пишет спецификацию, и те, кто ее читает, одни и те же слова и выражения понимают одинаково. Однако на самом деле это не так, поскольку естественному языку присуща определенная размытость понятий. Вследствие этого одно и то же требование может трактоваться разными людьми по-разному.

Чтобы избежать подобных проблем, разработаны методы описания требований, которые структурируют спецификацию и уменьшают размытость определений. Эти методы представлены в табл. 4.2. Кроме этого, разработаны другие подходы, например специальные языки описания требований.

Таблица 4.2. Способы записи спецификаций требований

Система записи	**Описание**
Структурированный естественный язык	Использование стандартных форм и шаблонов для написания естественный язык спецификации
Языки описания программ	Использование специальных структурированных языков, подобных языкам программирования, где спецификация требований строится на основе выбранной операционной модели системы
Графические нотации	Графический язык, использующий для описания функциональных требований диаграммы и блок-схемы, дополненные текстовыми пояснениями.
Математические спецификации	Это системы нотаций, основанные на математических концепциях, таких, как теория конечных автоматов или теория множеств.

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.