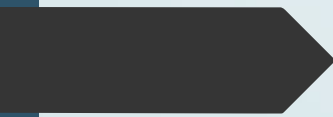




Создание спецификации требований к ПО



□ 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.

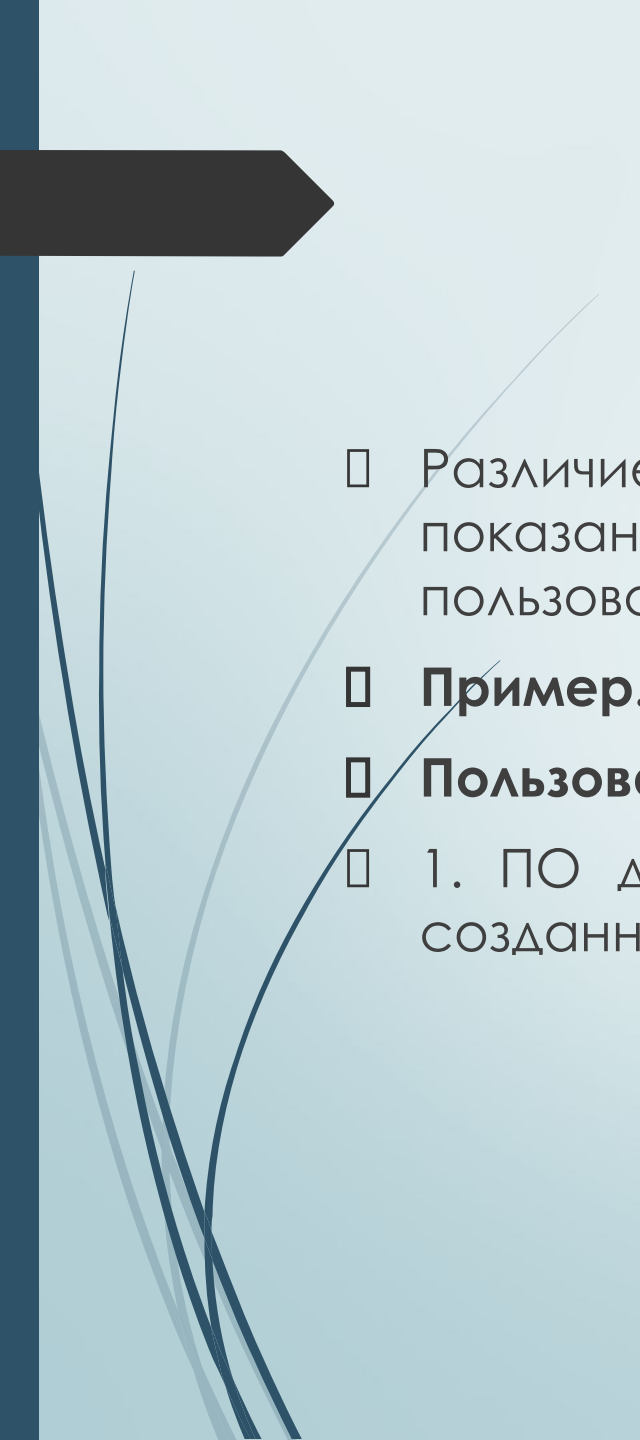
Например, если компания хочет выиграть контракт на разработку большого программного проекта, она вынуждена, пока решение не принято, представлять требования в самом обобщенном виде, чтобы, с одной стороны, удовлетворить требования заказчика, а с другой — иметь возможность для маневра при конкуренции 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 ее 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. Пользовательские и системные требования**

□ **Пользовательские требования**

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

□ Спецификация системных требований

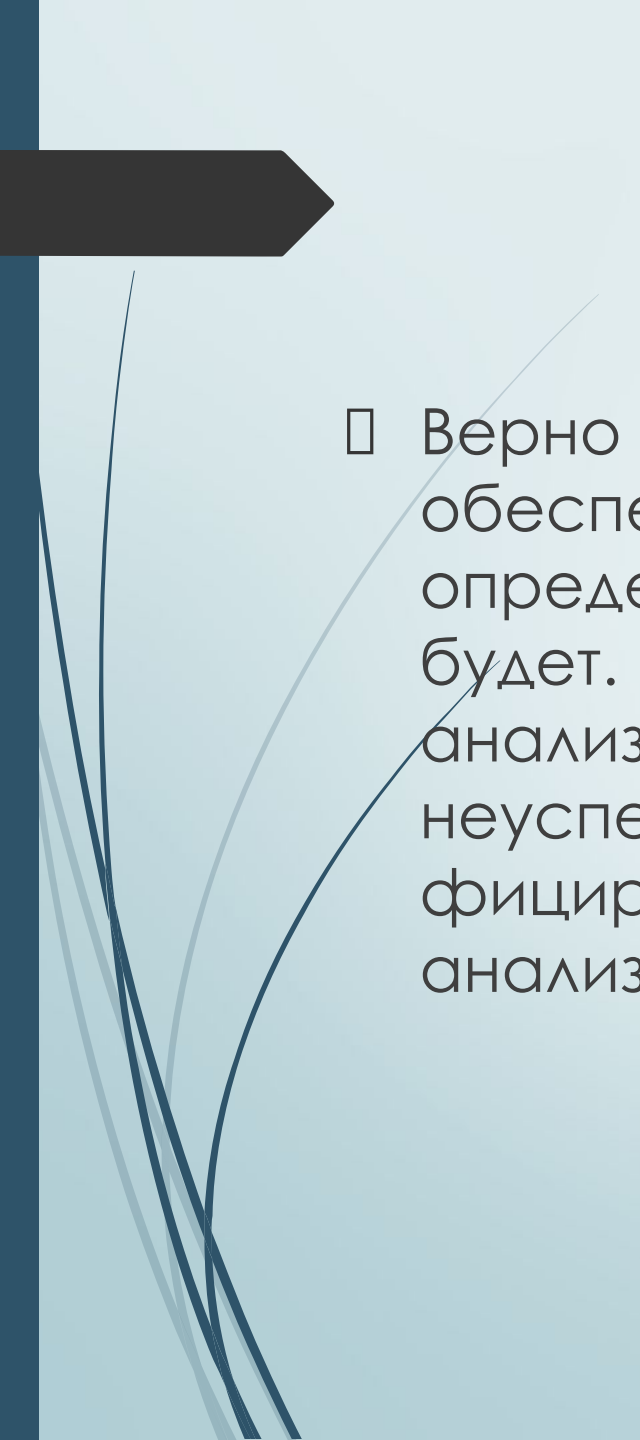
- Пользователь должен иметь возможность определять тип внешних файлов.
- Для каждого типа внешнего файла должно иметься соответствующее средство, применимое к этому типу файлов.
- Внешний файл каждого типа должен быть представлен соответствующей пиктограммой на дисплее пользователя.
- Пользователю должна 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.

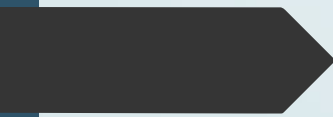


□ Выявление и анализ требований Фаза с таким или сходным названием (чаще всего применяют термин «разработка требований») присутствует во всех известных моделях жизненного цикла. Более того, современные тенденции в технологии программирования таковы, что данная фаза все чаще рассматривается не только как первая, но и как главная, а иногда и решающая фаза жизненного цикла. Дело в том, что заметные успехи технологии программирования позволяют, при наличии адекватных требований, организовать выполнение других фаз, если и не как автоматический, то, во всяком случае, как управляемый и измеримый процесс. Другими словами, если современные программисты точно знают, что именно нужно сделать, они это, скорее всего, сделают.



□ Верно и обратное: если требования к программному обеспечению не определены, или недостаточно определены, то, скорее всего, успешной разработка не будет. Об этом свидетельствует статистика, собранная при анализе проведения проектов — большая часть причин неуспешного выполнения конкретных проектов была квалифицирована как ошибки или недоработки на фазе анализа требований.

Схема разработки требований



Разработка требований - это первая из основных фаз процесса создания программных систем. Эта фаза состоит из следующих основных работ (рис. 5).

- **Анализ предметной области.** Позволяет выделить сущности предметной области, определить первоначальные требования к функциональности и определить границы проекта.
- **Анализ осуществимости.** Должен выполняться для новых программных систем. На основании анализа предметной области, общего описания системы и ее назначения принимается решение о продолжении или завершении проекта.

- 
- **Формирование и анализ требований.** Взаимодействуя с пользователями, обсуждая и анализируя с ними задачи, возлагаемые на систему, разрабатывая модели и прототипы, разработчики формулируют пользовательские требования.
 - **Документирование требований.** Сформированные на предыдущем этапе пользовательские требования должны быть документированы. При этом нужно учесть, что основными читателями этого документа будут пользователи, поэтому основными требованиями к нему будут ясность и понятность.
 - **Детализация требований.** Разработчики детализируют требования пользователей, формируя более точные подробные системные требования.



□ **Согласование и утверждение требований.** На этом этапе пользовательские и системные требования должны быть оформлены в виде единого документа, содержащего все функциональные и нефункциональные требования. Такой документ, обычно, называется спецификацией требований. Спецификация требований должна удовлетворять следующим характеристикам качества: корректность, однозначность, завершенность и согласованность.



□ На этапе анализа разрабатываются пользовательские и системные требования к программной системе, которые оформляются в виде единого документа - спецификации требований, - являющегося формальным соглашением заказчика с разработчиком системы. Практика показывает, что требования к разрабатываемой программной системе часто изменяются. Это обусловлено тем, что разработка программной системы довольно длительный процесс, во время которого:

□ понимание пользователями возможностей системы, решаемых ею задач, может измениться;

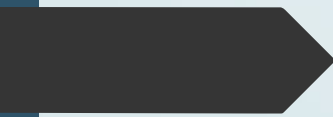
- 
- происходят изменения в деловой среде, техническом, программном и другом обеспечении системы, которые должны быть учтены;
 - понимание разработчиками поставленных перед ними задач меняется.
 - Под **управлением требованиями** понимают все действия, направленные на поддержание целостности, точности и актуальности спецификации требований в процессе разработки программной системы.

- 
- К действиям по управлению требованиями относятся:
 - определение основной (базовой) версии спецификации требований для конкретной версии продукта;
 - анализ предлагаемых изменений требований, оценка воздействия и стоимости каждого изменения до его принятия;
 - включение одобренных изменений при помощи определенной процедуры;
 - согласование плана проекта с требованиями;

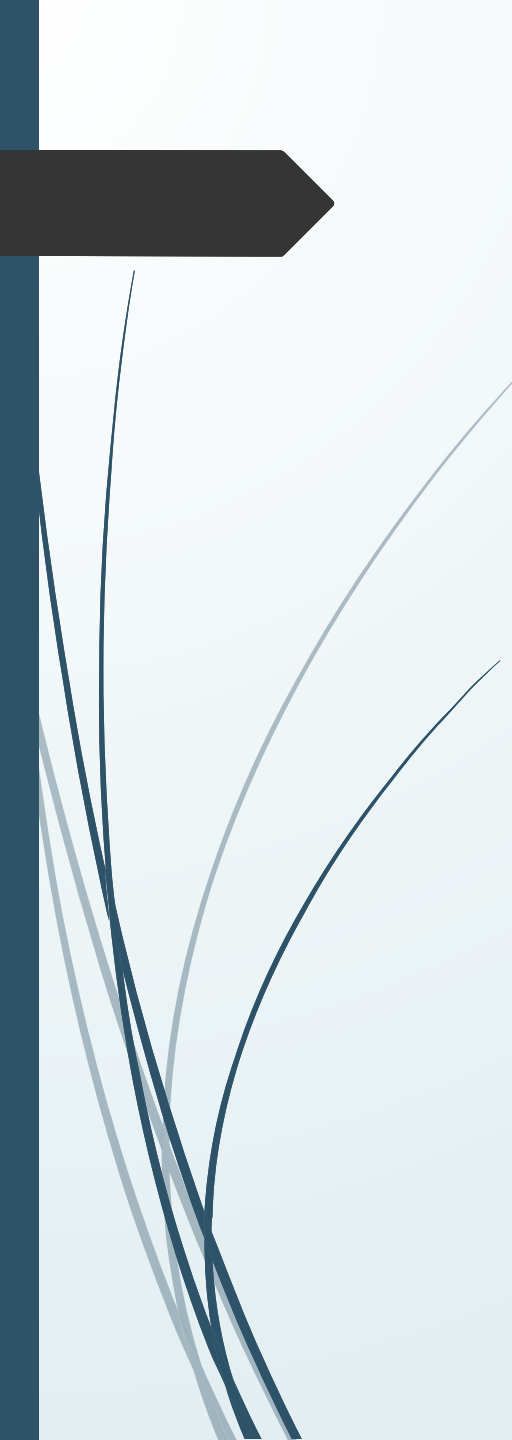
- 
- отслеживание отдельных требований от проектирования до кода приложения и его тестирования;
 - отслеживание статуса требований и действий по изменению на протяжении всего проекта.
 - В организации (или в проекте) должны быть определены действия по управлению требованиями. Эти действия должны быть документированы и должны выполняться всеми участниками проекта. При разработке процесса нужно определить:

- 
- методы и средства управления версиями спецификации и отдельных требований;
 - процесс разработки, согласования, экспертизы и утверждения базовой версии;
 - процесс присвоения, контроля и изменения статуса требования;
 - процесс передачи новых требований и изменений существующих требований заинтересованным лицам;
 - методы анализа влияния и стоимости внесения изменения.

- 
- Кроме этого описание процесса должно содержать определение участников проекта, ответственных за выполнение каждой конкретной задачи. Минимальной единицей управления в спецификации требований является отдельное требование, поэтому вопрос идентификации требования достаточно важен. Форма представления требования может быть различной (текстовая, графическая и т.д.), поэтому для идентификации требования обычно используют связанный с ним набор атрибутов. Атрибутами могут быть: дата создания требования, номер текущей версии требования, номер версии продукта, для которой предназначено требование, автор требования, ответственный за реализацию требования, состояние требования, происхождение и обоснование требования, подсистема, для которой предназначено требование и т.д.

- 
- Главное при выборе атрибутов, чтобы они однозначно идентифицировали требование и его состояние.
 - В процессе выполнения проекта требование, обычно, изменяет свое состояние от начального («предложено»), до конечного, например, «реализовано». Состояния требований позволяет оценить степень готовности проекта. Рекомендуются использовать состояния требования, приведенные в табл. 1.

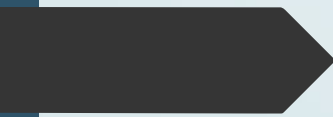
□ Таблица 1. Состояния требования



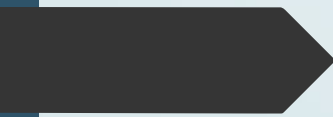
Состояние	Определение
Предложено(proposed)	Требование запрошено авторизированным источником
Одобрено (approved)	Требование проанализировано, его влияние на проект просчитано, и оно было размещено в базовой версии определенной версии продукта. Ключевые заинтересованные в проекте лица согласились с этим требованием, а разработчики обязались его реализовать.
Реализовано (implemented)	Код, реализующий требование разработан, написан и протестирован. Требование отслежено до соответствующих элементов дизайна и кода.
Проверено (verified)	Корректное функционирование реализованного требования подтверждено в соответствующем продукте. Требование отслежено до соответствующих вариантов тестирования. Теперь требование считается выполненным.
Удалено (deleted)	Утвержденное требование удалено из базовой версии. Следует зафиксировать причины и лицо, принявшее это решение.
Отклонено (rejected)	Требование предложено, но не запланировано для реализации ни в одной из будущих версий. Следует зафиксировать причины и лицо, принявшее это решение.

- 
- В процессе управления требованиями должны быть определены лица, которые могут изменить состояние требования. Управление статусом позволяет численно определить степень готовности проекта, считая, например, что основная часть работы закончена, если все требования имеют состояние «Проверено» или «Удалено».
 - После разработки, согласования и утверждения спецификация требований становится основным документом в проектировании системы (версии системы). Изменения в этот документ разрешается вносить только через определенный в организации (или проекте) процесс внесения изменений.

- 
- 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.



□ 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.

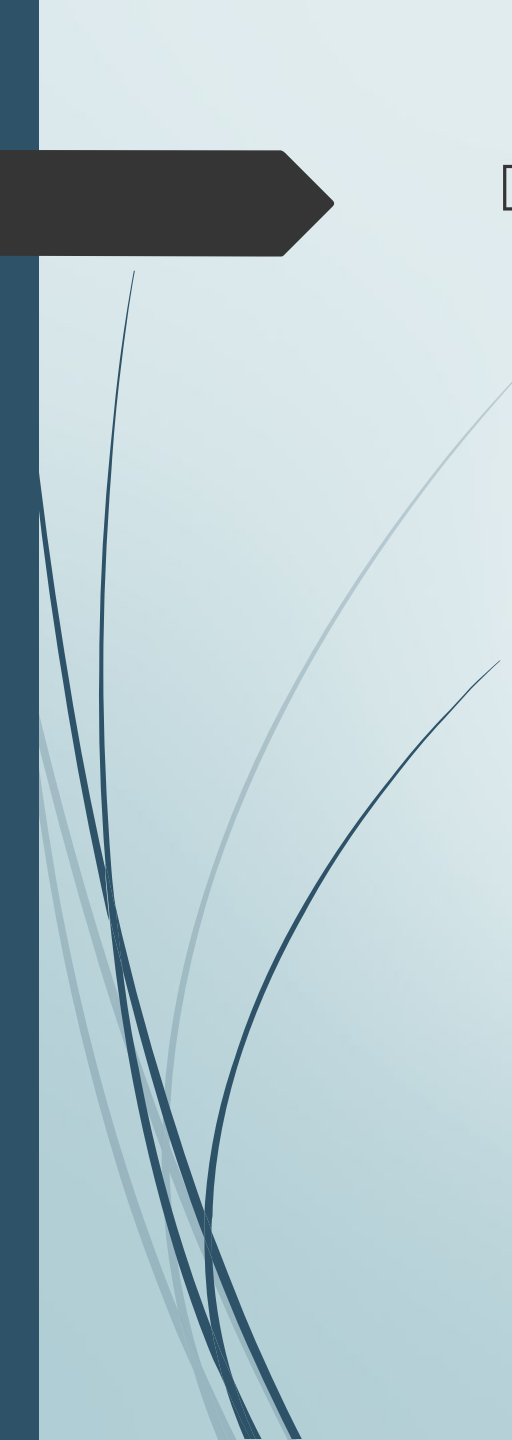
□ Группа нефункциональных требований (*Non-Functional Requirements*)

□ 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.