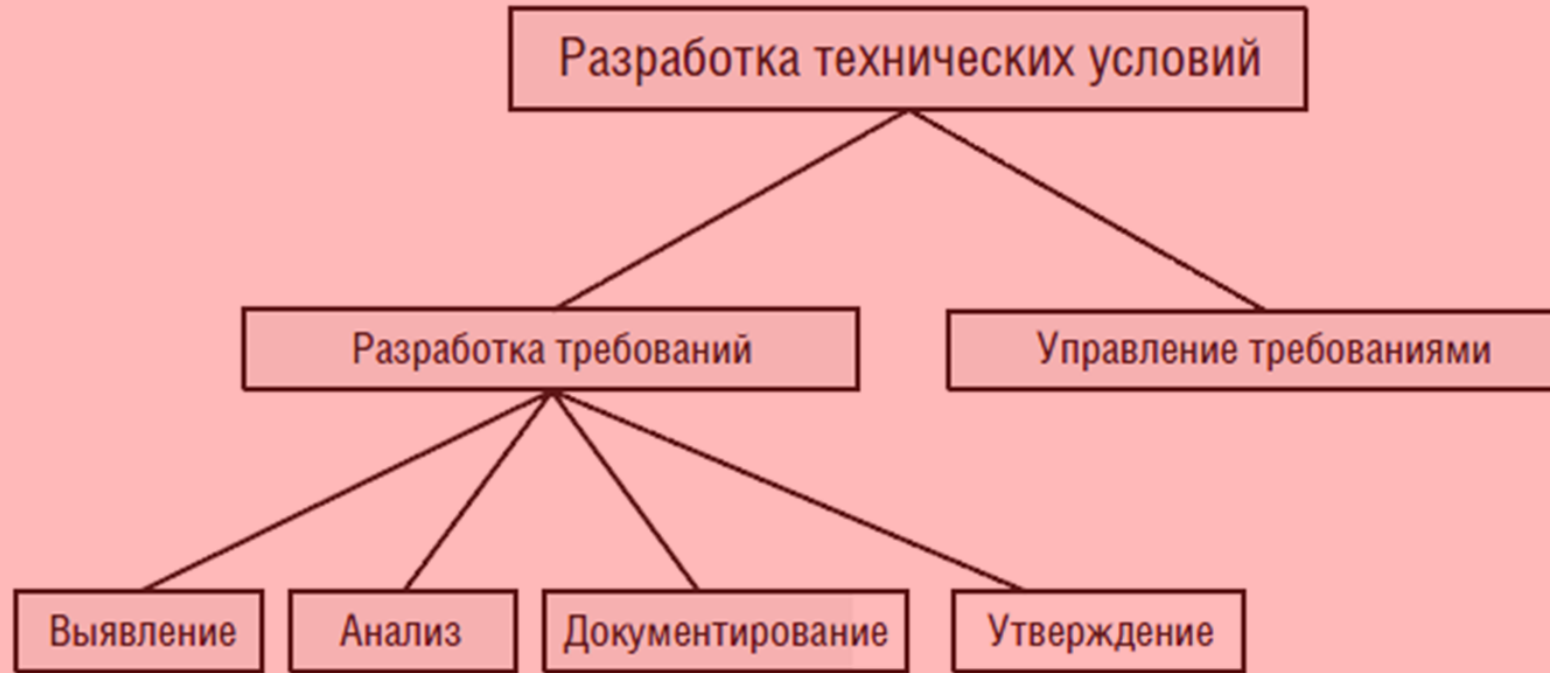


Тема 2

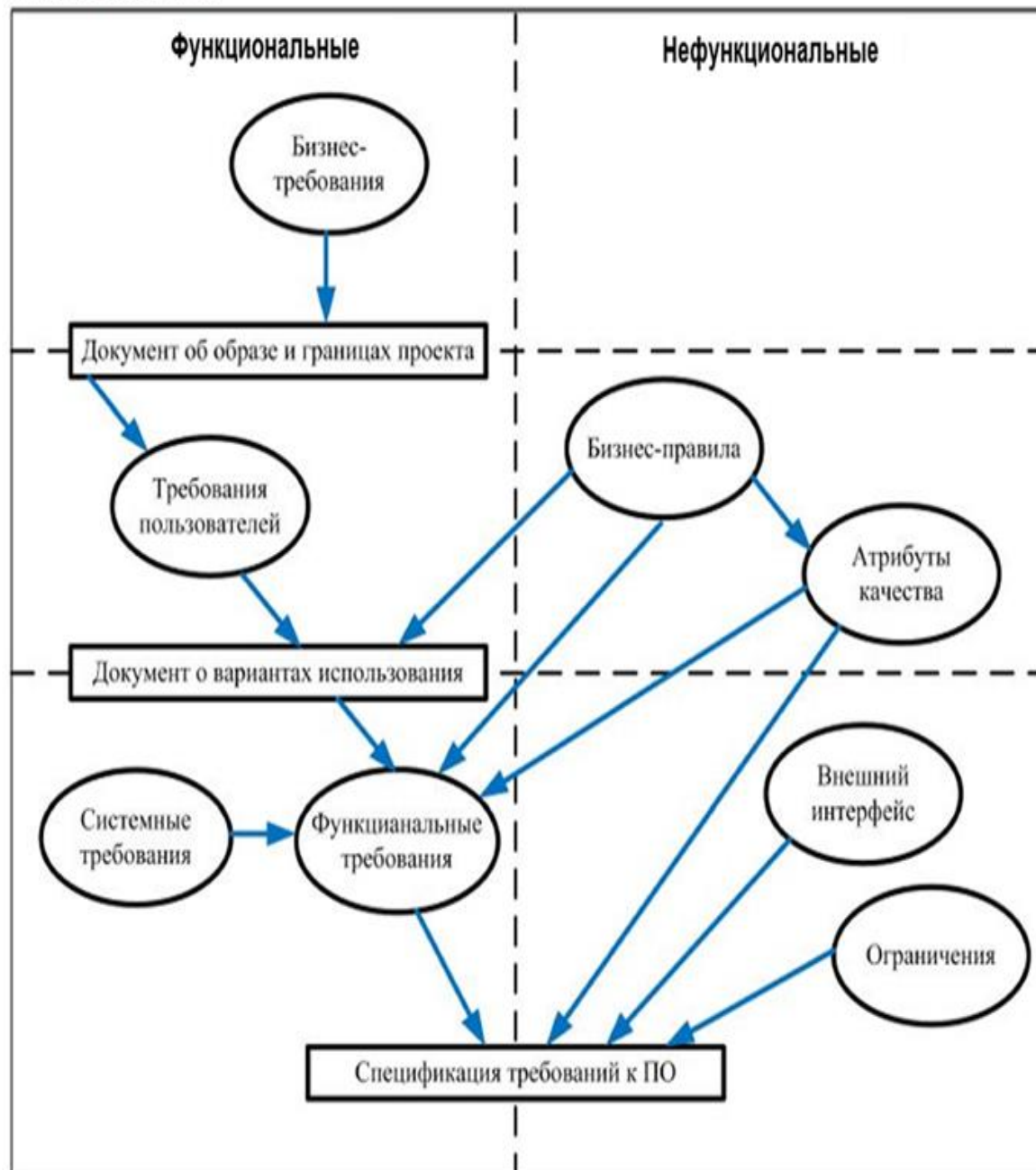
# **Принципы создания требований**

# Что изучаются в дисциплинах:

## Основы разработки и анализа требований к ПО.



Типы требований:



# 1. Идеальные требования

Не существует выверенного способа написания идеальных требований, а лучший учитель — это опыт, который нарабатывается со временем. Конструктивная критика проницательных коллег — огромное подспорье, потому что позволяет понять, когда ваш текст выполняет свою задачу, а когда нет.

Вот почему рецензирование документов требований коллегами так важно. Начните процесс рецензирования с дружбы с коллегой бизнес-аналитиком и взаимного рецензирования требований. Вы узнаете, как другой бизнес-аналитик пишет требования, и внесете свой вклад в успех команды за счет более раннего обнаружения ошибок и возможностей улучшения.

Говоря «**написание требований**», люди обычно подразумевают текстовое представление требований на естественном языке. Но лучше мысленно перефразировать «**написание требований**» на «**представление знания о требованиях**». Во многих случаях альтернативные варианты представления информации доводят ее более эффективно, чем простой текст.

Бизнес-аналитик должен выбрать подходящее сочетание методов коммуникации, которые гарантируют ясное и единое понимание как потребностей заинтересованного лица, так их решения, которое планируется создать.

При написании требований преследуются две важных цели:

- любой, прочитавший требование, должен понимать его в той же интерпретации, что и любой другой читатель этого требования;
- интерпретации всех читателей требования должны совпадать с тем, что автор хотел передать.

Эти результаты важнее, чем чистота стиля и догматическое следование каким-то заданными правилам и соглашениям.

## 2. Системная или пользовательская точка зрения

Функциональные требования можно писать с точки зрения того, **что делает система или что делает пользователь**. Так как важнее всего эффективно передать информацию, нет ничего страшного в смешении этих подходов и формулировании каждого требования в максимально понятной форме.

**Требования следует излагать последовательно**, например «**Система должна**» или «**Пользователь должен**», затем — **активный глагол**, а после — **наблюдаемый результат**. Укажите иницирующие условия или триггеры, вследствие которых система ведет себя определенным образом.

Общий шаблон требования, написанного с точки зрения системы (см. «Easy Approach to Requirements Syntax (EARS)»):

[необязательное предварительное условие] [необязательный триггер события]  
**система должна** [ожидаемая реакция системы].

Пример простого функционального требования, описывающего действие системы, на основе этого шаблона:

*«Если запрошенный химикат есть на складе химикатов, **система должна** отобразить список всех хранимых на складе контейнеров с указанным химикатом».*

Этот пример содержит предварительное условие, но в нем нет триггера. Некоторые авторы требований опускают из требований фразу «**система должна**», аргументируя это тем, что поскольку требования описывают поведение системы, нет необходимости постоянно упоминать, что «**система должна**».

В данном примере удаление фразы «система должна» не изменит смысла.

Иногда естественнее выразить требование в терминах действий пользователя, а не с точки зрения системы. Включение «**должен**» и использование активного залога позволяет четко сказать, какая сущность выполняет действие.

При написании функциональных требований с точки зрения пользователя хорошо работает следующая общая структура (шаблон):

*[класс пользователя или имя действующего лица] должен иметь возможность [выполнить что-то] [с каким-то объектом] [условия выполнения, время отклика или декларация качества].*

Альтернативная формулировка шаблона выглядит так:

«Система должна позволять (или разрешать или давать возможность) [название класса пользователя] [делать что-то].»

Пример функционального требования, написанного с точки зрения пользователя:

*Химик должен иметь возможность повторно заказать любой химикат, который он ранее заказывал, путем извлечения и редактирования параметров ранее введенного заказа.*

В требовании используется название класса пользователя — **Химик**, а не общий термин «**пользователь**». Явная формулировка максимально снижает вероятность неверного истолкования.



### 3. Язык и стиль

Написание требований не похоже на создание художественных и нехудожественных произведений. В данном случае не подходит стиль письма, к которому вы привыкли в школе и институте и в котором *сначала излагается основная идея, потом описываются подкрепляющие основную идею факты и, наконец, формулируется заключение.*

Надо изменить стиль письма: *первой формулируется суть — формулировка потребности или функциональности, после чего следуют детали (обоснование, источник, приоритет и другие атрибуты требования).* Такая структура удобная как читателям, которые бегло просматривают документ, так и тем, кому нужны все детали.

Включение таблиц, структурированных списков, диаграмм и других визуальных элементов помогает оживить монотонное перечисление функциональных требований и предоставляет более богатую информацию тем, кому проще осваивать материал нетрадиционными методами.

### 3.1. Ясность и лаконичность

Пишите требования полными предложениями, с правильной грамматикой, правописанием и пунктуацией. Предложения и абзацы должны быть краткими и ясными. Пишите требования простым и прямолинейным языком, характерным для соответствующей предметной области, но не используйте профессиональный жаргон. *Определения используемых терминов размещайте в словаре терминов.*

Еще одна хорошая привычка — писать кратко. Фразы типа «должна предоставлять пользователю возможность сделать» можно свести к одному слову — «**должна**».

Для каждой включаемой в требования порции информации задавайте себе вопрос: «Что читатель будет делать с этой информацией?». Если вы не уверены, что ваша информация будет полезной каким-либо заинтересованным лицам, скорее всего она не нужна.

*Вместе с тем ясность важнее лаконичности.*

### 3.2. Ключевое слово «должна»

Традиционно при описании какой-либо возможности системы используют слово «**должна**». Формулировки со словом «**должна**» четко говорят о требуемой функциональности — в полном соответствии с самой важной целью ясного и четкого доведения информации. Вам может больше понравиться другие слова, такие как «**обязана**», «**следует**» или что-то похожее, но **нужно придерживаться одного термина**.

Иногда при написании спецификаций используют произвольные и вводящие в заблуждения наборы глаголов: *может, будет, надлежит, должна бы, могла бы, обязана, надо бы, должна бы обеспечить и т. п.*

Не всегда понятно, есть различия в значении этих глаголов или нет. Нюансы смысла различных глаголов также сильно усложняют единообразную интерпретацию документа в командах, члены которых относятся к разным культурам. Лучше всего придерживаться ключевого слова «**должна**».

### 3.3. Активный залог

Используйте активный залог, чтобы дать точно понять, какая сущность выполняет описываемое действие. Много научных и бизнес-текстов написаны с использованием пассивного залога, но он никогда не бывает таким ясным и прямым, как активный залог.

Следующее требование написано с использованием пассивного залога:

*После доставки обновления продукта **будет изменен** регистрационный номер в соответствующей строке договора.*

Фраза «**будет изменен**» — признак пассивного залога. Она обозначает адресата (регистрационный номер), но не исполнителя действия. То есть эта фраза ничего не говорит о том, кто или что обновляет регистрационный номер. Выполняет ли это система автоматически или регистрационный номер должен обновить пользователь? Изменение формулировки этого требования с использованием активного залога явно показывает действующее лицо и уточняет иницирующее событие:

*Когда отдел обслуживания подтвердит, что они отправили обновление продукта, **система должна** обновить клиентский договор, задав в нем новый регистрационный номер продукта.*

### 3.4. Раздельные требования

Избегайте длинных повествовательных абзацев, которые содержат несколько требований. Читатели не должны вылавливать изюминки отдельных требований из единой безликой массы текста. Четко отделяйте требования от вводной и контекстной информации. Такая информация полезна читателям, но они должны четко отличать формулировки требований.

Наличие в требовании таких слов, как *«и»*, *«или»*, *«дополнительно»* и *«также»*, предполагает, что несколько требований могли быть объединены.

Это не означает, что нельзя использовать союз «и», но если вы делаете это, проверяйте, соединяет ли он две части одного требования или два отдельных требования. **Если для проверки двух частей требуются разные тесты, разбейте предложение на отдельные требования.**

**Никогда не используйте «и/или» в требованиях; это оставляет читателю свободу интерпретации, вот пример:**

*Система должна поддерживать поиск по номеру заказа, номеру счета и/или по номеру пользовательского заказа на покупку.*

Такое требование позволило бы пользователю ввести одно, два или три числа при выполнении одной операции поиска. Скорее всего, это не то, что планировалось.

Такие выражения, как «**пока не**» и «**кроме**» также указывают на наличие нескольких требований:

*Кредитная карточка покупателя должна считаться действительной для платежей **до тех пор, пока не** истечет ее срок действия.*

Отсутствие информации о том, что происходит, когда выполняется условие «**пока не**», — обычная причина наличия недостающих требований. Разделите это положение на два — для двух условий: когда кредитная карточка действительна и когда срок ее действия истек:

*Если кредитная карточка покупателя действительна, **система должна** выполнить платеж по этой карточке.*

и

*Если срок действия кредитной карточки покупателя истек, **система должна** предоставить покупателю возможность обновить информацию своей кредитной карты или ввести для платежа информацию другой кредитной карточки.*

### 3.5. Неоднозначные слова

Последовательно используйте термины и именно так, как они определены в словаре. Остерегайтесь синонимов и слов, близких по значению. Не используйте разные термины для обозначения одного элемента в одном и том же документе требований. Выберите один термин и последовательно используйте только его, а синонимы укажите в словаре терминов, чтобы люди, привыкшие к другому названию, могли связать его с вашим термином.

При использовании местоимения, ссылающегося на что-то упомянутое ранее, обеспечьте, чтобы было четко понятно, на что вы ссылаетесь.

Наречия привносят субъективность, а, значит, неоднозначность. Избегайте таких слов, как *разумно, уместно, вообще, приблизительно, обычно, систематично и быстро*, потому что нет уверенности, что читатель поймет их правильно.

Требования, изложенные неясным языком, не поддаются проверке, поэтому избегайте двусмысленных и субъективных терминов. Некоторые из этих слов могут применяться в бизнес-требованиях, но только не в пользовательских или конкретных функциональных требованиях, описывающих решение, которое планируется создать.



### 3.6. Некоторые неоднозначные термины, которых следует избегать в спецификации к требованиям

Неоднозначные термины	Способы улучшения
Приемлемый, адекватный	Определите, что понимается под приемлемостью и как система это может оценить
И/или	Укажите точно, что имеется в виду — «и» или «или», чтобы не заставлять читателя гадать
Практически выполнимо	Не заставляйте разработчиков определять, что под этим понимается. Поставьте пометку «TBD» и определите дату, к которой эту проблему следует разрешить
По меньшей мере, как минимум, не более чем, не должно превышать	Укажите минимальное и максимальное допустимые значения
Наилучший, самый больший, большинство	Укажите, какой уровень требуется, а также минимальный приемлемый уровень
Между, от X до Y	Укажите, входят ли конечные точки в диапазон



Неоднозначные термины	Способы улучшения
Зависит от	Определите природу зависимости. Обеспечивает ли другая система ввод данных в вашу систему, надо ли установить другое ПО до запуска вашей системы и зависит ли ваша система от другой при выполнении определенных расчетов или служб?
Эффективный	Определите, насколько эффективно система использует ресурсы, насколько быстро она выполняет определенные операции и как быстро пользователи с ее помощью могут выполнять определенные задачи
Быстрый, скорый, моментальный	Укажите минимальное приемлемое время, за которое система выполняет определенное действие
Гибкий, универсальный	Опишите способы адаптации системы в ответ на изменения условий работы, платформ или бизнес-потребностей
Улучшенный, лучший, более быстрый, превосходящий, более качественный	Определите количественно, насколько лучше или быстрее должны стать показатели в определенной функциональной области или аспект качества

Неоднозначные термины	Способы улучшения
Включает; включает в себя, но не ограничен этим; и т. д.; и т. п.; такой как, в частности	Перечислите все возможные значения или функции, а не только примеры, или укажите читателю, где можно найти исчерпывающий список. В противном случае разные читатели могут по-разному понимать, что должен содержать полный список и где он должен заканчиваться
В большинстве случаев, обычно, как правило, практически всегда	Уточните, когда указанные условия или сценарии неприменимы и что должно происходить в таком случае. Опишите, как пользователь или система должны различать разные случаи
Соответствует, равняется, согласуется, представляет то же самое	Определите, должно ли сравнение текста быть чувствительным к регистру и что означает эта фраза — «содержит», «начинается с» или «точно совпадает». Для действительных чисел определите точность при сравнении
Максимизируйте, минимизируйте, оптимизируйте	Укажите минимальное и максимальное допустимые значения определенного параметра
Обычно, в идеальном варианте	Опишите нештатные или неидеальные условия и как система должна вести себя в таких ситуациях

Неоднозначные термины	Способы улучшения
Необязательно	Укажите, кто делает выбор: система, пользователь или разработчик
Возможно, желательно, должно	Должно или не должно?
Разумный, при необходимости, когда уместно, по возможности	Объясните четко, как разработчик или пользователь должен оценивать разумность и уместность
Устойчивый к сбоям	Определите, как система должны обрабатывать исключения и реагировать на неожиданные условия работы
Цельный, прозрачный, корректный	Что означает «цельный» или «корректный» для пользователя? Выразите ожидания пользователя, применяя характеристики продукта, которые можно наблюдать
Несколько, некоторые, много, немного, множественный	Укажите сколько или задайте минимальную и максимальную границы диапазона
Не следует	Старайтесь формулировать требования в позитивной форме, описывая, что именно система будет делать
Современный	Поясните этот термин для заинтересованного лица
Достаточный	Укажите, какая степень чего-либо свидетельствует о достаточности
Поддерживает, позволяет	Дайте точное определение, из каких действий системы состоит «выполнение» конкретной возможности
Дружественный, простой, легкий	Опишите системные характеристики, которые будут отвечать потребностям пользователей и его ожиданиям, касающимся легкости и простоты использования продукта

### 3.7. Конструкция типа «А/В»

Многие спецификации требований содержат выражения вида «А/В», то есть двух связанных терминов (синонимов или антонимов) разделенных косой чертой. Такие выражения как правило неоднозначны. Вот пример:

*Система должна обеспечивать автоматический сбор информации о лицензионных ключах при массовом выпуске продукта отделом доставки/исполнения.*

Это предложение можно истолковать несколькими способами:

- «отдел доставки/исполнения» — это название подразделения;
- доставка и исполнение являются синонимами;
- в некоторых проектах упомянутое подразделение называется отделом доставки, а в других — отделом исполнения;
- массовый выпуск продукта может выполнять отдел доставки или отдел исполнения, так что косая черта означает «или»;
- массовый выпуск продукта отдел доставки или отдел исполнения выполняют совместно, так что косая черта означает «и».

**Лучше точно решить, что хочется сказать, и выразить это правильными словами.**

### 3.8. Пограничные значения

Много неоднозначности возникает на границах числовых диапазонов как в требованиях, так и бизнес-правилах. Пример:

*Отпуск длительностью до 5 дней не требует одобрения. Запросы на отпуск длительностью от 5 до 10 дней требуют одобрения непосредственного начальника. Отпуска длительностью 10 дней и более требуют одобрения директора.*

При такой формулировке непонятно, в какую категорию попадают отпуска длительностью точно 5 и 10 дней. Если ввести дробные числа, скажем 5,5 дней отпуска, то дело еще больше запутается.

Слова «от и до», «включительно» и «свыше» вносят четкость и ясность на счет пограничных значений:

*Отпуск длительностью 5 дней и меньше не требует одобрения. Запросы на отпуск длительностью более 5 и до 10 дней включительно требуют одобрения непосредственного начальника. Отпуска длительностью свыше 10 дней требуют одобрения директора.*

### 3.9. Негативные требования

Иногда в требованиях пишут не то, *система должна*, а то, что она *не должна делать*. Как реализовать такие «негативные» требования? Особенно сложны в расшифровке двойные и тройные отрицания.

Попробуйте переформулировать негативные требования в позитивном стиле, которые описывает ограничение на поведение.

Пример:

*Пользователь не должен иметь возможность активизировать договор, если он не сбалансирован.*

Лучше перефразировать это двойное отрицание («не должен» и «не сбалансирован») как позитивное утверждение:

*Система должна позволять пользователю активировать договор, только если этот договор сбалансирован.*