

Введение в бизнес-анализ

Выявление бизнес-требований и бизнес-правил



На этом уроке

1. Определим различия бизнес-целей, бизнес-правил и бизнес-требований.
2. Классифицируем бизнес-правила и разберёмся со способами их выявления.
3. Рассмотрим основные виды бизнес-требований и методики их сбора.
4. Познакомимся с основными этапами разработки требований.
5. Узнаем об управлении требованиями и управлении проектом.

Оглавление

[Типы бизнес-правил](#)

[Уровни бизнес-требований](#)

[Методы сбора требований](#)

[Анкетирование](#)

[Интервью](#)

[Автозапись](#)

[Изучение документации](#)

[Повторное использование документации](#)

[Представитель заказчика в компании разработчика](#)

[Работа в поле](#)

[Обучение](#)

[Мозговой штурм](#)

[Совещание](#)

[Use cases](#)

[Этапы разработки требований](#)

[Выявление требований](#)

[Анализ требований](#)

[Спецификации требований](#)

[Проверка требований](#)

[Управление требованиями](#)

[Управление требованиями в рамках управления проектом](#)

[Практическое задание](#)

[Глоссарий](#)

[Используемые источники](#)

Определение цели бизнеса — это то, с чего начинается каждое дело, будь то организация небольшой фирмы или же многомиллионный проект. Считается, что цель бизнеса во всех случаях очевидна и заключается она в получении прибыли. Перед началом проекта необходимо определить бизнес-цели, а также бизнес-правила и бизнес-требования.

Чтобы разобраться с данными понятиями и их различиями, вспомним определения из первых двух уроков курса.

Бизнес-правило — это конкретная, исполняемая и проверяемая директива, которая находится под контролем организации и служит для поддержания бизнес-политики.

Бизнес-требования — объем информации, который в совокупности описывает потребность, которая инициирует один или больше проектов, призванных предоставить решение и получить требуемый конечный бизнес-результат. Бизнес-требования включают бизнес-возможности, бизнес-цели, метрики успеха, концепция и границы и ограничения.

Бизнес-цель — это финансовая или нефинансовая выгода, которую организация ожидает получить в результате реализации проекта или другой инициативы.

Отличия бизнес-правил, бизнес-требований и бизнес-целей приведены в таблице.

Бизнес-правило	Бизнес-требование	Бизнес-цель
Сочетание принципов и выводов, которое определяет, как мы делаем бизнес	Включают бизнес-возможности, бизнес-цели, метрики успеха, концепция и границы и ограничения	Выгода для организации, которую она получит после внедрения нового решения

Типы бизнес-правил

Назначение бизнес-правил — защищать структуру бизнеса, контролировать его операции или влиять на них.

В настоящее время существует ряд методологий, разработанных специально для создания и документирования бизнес-правил и их использования при разработке автоматизированных информационных систем. Однако на практике зачастую достаточно выявить и задокументировать относящиеся к системе правила и связать их с конкретными функциональными требованиями к приложению.

Для организации бизнес-правил предлагается множество различных схем классификации. Простейшая из них приведена на рисунке. Она состоит из пяти типов бизнес-правил:



Вид правил	Определение	Пример
Факты (Facts)	<p>Это верные утверждения о бизнесе. Они описывают связи и отношения между важными бизнес-терминами. Факты также называют инвариантами — неизменными истинами о сущности данных и их атрибутах.</p> <p>Обычно факты не преобразуются напрямую в функциональные требования к системе.</p> <p>Сведения о сущности данных, важных для системы, применяют в моделях данных, создаваемых аналитиком или архитектором базы данных</p>	<p>На каждый контейнер с реактивом нанесён уникальный штрихкод.</p> <p>Заказ должен содержать не менее одной позиции.</p> <p>Каждый авиарейс имеет аэропорт вылета и аэропорт прилёта</p>
Ограничения (Constraints)	<p>Определяют, какие операции могут выполняться в рамках системы. Как правило, при формулировании ограничений используются слова и фразы вида «должен / не должен», «может / не может», «только»</p>	<p>Сотрудник может запросить вещество из списка химикатов первого уровня опасности, только если за последние 12 месяцев он прошёл обучающий курс по работе с опасными соединениями.</p> <p>Экипажи коммерческих авиарейсов должны каждые 24 часа отдыхать не менее 8 часов</p>
Активаторы операций (Action enabler)	<p>Правила, при определённых условиях приводящие к выполнению каких-либо действий. Выражение вида «Если <некоторое условие верно или наступило определённое событие>, то <что-то произойдёт>», — это ключ, который описывает активатор операции</p>	<p>Если срок хранения контейнера с химикатом истёк, необходимо уведомить лицо, у которого в данный момент находится контейнер.</p> <p>Если на запрошенную клиентом дату вылета рейс не выполняется,</p>

		необходимо предоставить ему расписание вылетов по данному направлению
Вывод (Inference)	Это правило, устанавливающее новые реалии на основе достоверности определённых условий. Вывод создаёт новый факт на основе других фактов или вычислений. Выводы зачастую записывают в формате «если — то», применяемом при записи активаторов. Однако раздел «то» вывода заключает в себе факт или предположение, а не действие	<p>Считается, что срок хранения контейнеров с химикатами, разлагающимися на взрывоопасные составляющие, истекает через один год с даты изготовления.</p> <p>Если в расписании зимнего сезона отсутствуют сведения о рейсе, выполняемом в летний период, рейс считается отменённым на весь период действия нового расписания</p>
Вычисления (Computations)	Вид бизнес-правил, определяющий вычисления, которые выполняются с использованием математических формул и алгоритмов. В отличие от активирующих операций, вычисления — это операции бизнес-правил, для реализации которых иногда приходится создавать специфические функциональные требования. Правила вычислений в той форме, в которой они выражены, можно рассматривать в качестве требований к программному обеспечению	<p>Цена единицы товара снижается на 10% при заказе от 6 до 10 единиц, на 20% — при заказе от 11 до 20 единиц и на 30% — при заказе свыше 20 единиц.</p> <p>Стоимость автомобиля Opel в салоне рассчитывается путём сложения стоимости базовой модели, стоимости выбранных покупателем пакетов и стоимости опций за минусом скидки при повторной покупке в салоне и скидки корпоративного клиента. Но при условии, что продаваемый автомобиль не реализуется по акции «КАСКО в подарок»</p>

Рекомендуется писать бизнес-правила на элементарном уровне, а не объединять множество деталей. Сложное правило может ссылаться на используемые в нём отдельные правила. Это делает их

краткими и простыми. Кроме того, появляется возможность их повторного использования и создания различных комбинаций.

Для описания вычислений и активирующих операции бизнес-правил на элементарном уровне не стоит использовать в левой части конструкции «если — то» логику «или», а в правой части этой же конструкции — логику «и».

К элементарным бизнес-правилам, которые влияют на вычисление общей стоимости автомобиля из примера выше, относятся следующие:

- если клиент совершает повторную покупку, стоимость автомобиля уменьшается пропорционально скидке постоянного клиента;
- для корпоративных клиентов стоимость автомобиля уменьшается пропорционально скидке корпоративного клиента;
- скидки корпоративного и постоянного клиента суммируются.

Эти бизнес-правила называются элементарными, потому что дальнейшая их детализация невозможна. В итоге, скорее всего, будет создано множество элементарных бизнес-правил, от комбинаций которых будут зависеть все вычисления и функциональные требования.

Уровни бизнес-требований

Как мы помним из первых двух уроков,

Требование — это любое условие, которому должна соответствовать разрабатываемая система или программное средство. Требованием может быть возможность, которой система должна обладать, и ограничение, которому система должна удовлетворять.

IEEE Standard Glossary of Software Engineering Terminology (1990) определяет требования так:

1. Условия или возможности, необходимые пользователю для решения проблем или достижения целей.
2. Условия или возможности, которыми должна обладать система или системные компоненты, чтобы выполнить контракт или удовлетворять стандартам, спецификациям или другим формальным документам.
3. Документированное представление условий или возможностей для пунктов 1 и 2. Это определение охватывает требования пользователей, от которых зависит внешнее поведение системы, и разработчиков — они определяют некоторые скрытые параметры. Термин «пользователи» следует распространить на всех заинтересованных лиц, так как не все, кто заинтересован в проекте — пользователи.

Таким образом, **требования** — это спецификация того, что должно быть реализовано. В них описано поведение системы, свойства системы или её атрибуты. Они могут быть ограничены процессом разработки системы.

Проще говоря, **требование** — это зафиксированное желание пользователя, которое должна выполнять система. Но это определение не идеально, потому что есть требования, которые пользователь не высказывает явно. Также пользователь может высказать какое-нибудь пожелание, но оно не будет реализовано, либо требование может быть высказано в устной форме. Рекомендуется принять за правило, что если требование не зафиксировано в письменном виде, то оно не существует.

Три основных уровня требований к ПО:

1. **Бизнес-требования** — что система должна делать с точки зрения бизнеса. Понятие «бизнес» в данном контексте близко к понятию «заказчик». Пример бизнес-требования: промосайт, привлекающий внимание определённой аудитории к определённой продукции компании.
2. **Требования пользователей** описывают цели и задачи пользователей системы, которые должны достигаться или выполняться ими при помощи создаваемой программной системы. Эти требования часто представляют в виде вариантов использования (use cases). Иначе говоря, пользовательские требования — это то, что может сделать пользователь: зарегистрироваться, посмотреть определённую информацию, пересчитать данные по определённому алгоритму и прочее.
3. **Функциональные требования** определяют функциональность (поведение) программной системы, которая должна быть создана разработчиками, чтобы пользователи могли выполнять свои обязанности в рамках бизнес-требований и в контексте пользовательских требований. Другими словами, что будут делать разработчики, чтобы выполнить пользовательские требования.

Методы сбора требований

Сбор требований — это один из самых важных этапов создания любой информационной системы, будь то десктопное, веб- или мобильное приложение или же просто доработка уже существующего решения. До начала сбора требований должны быть выявлены все заинтересованные лица или стейкхолдеры, которые будут пользоваться системой. Чем точнее будет этот список, тем полнее будут требования.

Вспомним, кто такие стейкхолдеры.

Стейкхолдеры (stakeholders) — лица или организации, имеющие права, долю, требования или интересы относительно системы или её свойств, удовлетворяющих их потребностям и ожиданиям.

Стейкхолдерами могут быть любые физические лица и/или организации, которые активно участвуют в проекте и чьи интересы могут быть затронуты не только в процессе создания системы, но и непосредственно по завершении самого проекта. Ими могут быть менеджеры, начальники отделов, директора, любые сотрудники организации, которые будут хоть как-то взаимодействовать с готовым решением и чьи требования, пожелания, идеи, потребности, проблемы необходимо учесть.

Существует множество различных техник сбора требований, которые позволят лучше понять, чего же хочет заказчик. Комбинирование методик позволяет повысить эффективность сбора требований, а также избежать их потери. Тщательно собранные требования минимизируют риски проекта, так как позволяют сформировать чёткий и понятный базис для разработки системы. При сборе требований необходимо помнить, что важны не только функциональные требования — **что** делает система, но и нефункциональные — **как** система это делает. Рассмотрим основные методы сбора требований.

Анкетирование

Анкетирование — это составление листа-опросника (анкеты, брифа), который может содержать открытые вопросы, когда требуется сформулировать ответ, и закрытые, когда требуется выбрать ответ из предложенных вариантов.

Анкетирование используется, чтобы подтвердить или детализировать ранее известные требования, выбрать параметры для решений. Самый известный пример анкетирования — бриф на разработку сайта. Это анкета, содержащая список основных требований к будущему сайту.

Преимущества	Недостатки
<ol style="list-style-type: none">1. Высокая скорость получения результатов.2. Сравнительно небольшие материальные затраты	<ol style="list-style-type: none">1. Не подходит для выявления неявных требований.2. При составлении опросника физически невозможно учесть все необходимые вопросы

Интервью

Интервью — это своего рода беседа по душам, тет-а-тет с заинтересованным лицом.

В процессе интервью необходимо задавать открытые вопросы, которые служат для получения информации, и закрытые, которые нужны, чтобы подтвердить или опровергнуть конкретные варианты

требований. Интервью применяется в основном для получения информации по какой-либо конкретной теме и/или для уточнения требований.

Многим этот способ может показаться достаточно лёгким, но это не так. Провести хорошее интервью сложно. Необходимо гибко реагировать на реакцию интервьюируемого и в случае необходимости изменять порядок заготовленных вопросов или их формулировку. Также полезно включить диктофон во время интервью или вести заметки.

Преимущества	Недостатки
<ol style="list-style-type: none">1. Возможность задавать вопросы в произвольной последовательности.2. Возможность использовать вспомогательный материал.3. Возможность наблюдать невербальные реакции опрашиваемого человека и делать выводы о достоверности его ответов	<ol style="list-style-type: none">1. Интервью отнимает достаточно много времени и сил.2. Можно получить одинаковые ответы от интервьюируемых

Автозапись

Автозапись — это работа с записями, письмами (электронными письмами), а также с любыми другими документами, автор которых заказчик или конечный пользователь, то есть стейкхолдер.

В действительности это может быть и документ, и записанная на диктофон последовательность действий, и самый обычный черновик, где заказчик набросал свои идеи, проблемы, хотелки, которые нужно превратить в полноценные требования, согласовать и передать в разработку.

Пример: работа с концепцией или видением проекта, которое он прислал вам на момент начала работ по аналитике. Заказчики любят называть это «ТЗ».

Преимущества	Недостатки
Помогает лучше понять сложные процедуры или процессы	Результат сильно зависит от опыта заказчика, а также от его умения формулировать и выражать свои мысли

Изучение документации

Изучение документации используется при наличии в организации документации, которая может помочь в определении потребностей заказчика.

Примеры документации: регламенты, описания бизнес-процессов, структуры организации, спецификации продукта, различные процедуры, стандарты и инструкции, шаблоны документов и т. д.

Выявленные с помощью изучения документации требования — это основа для дальнейшего анализа, и они должны быть детализированы. Данная методика применима, например, при автоматизации устоявшихся в организации регламентированных бизнес-процессов.

Преимущества	Недостатки
Быстрое получение информации	Не подходит при наличии в компании только базовых документов и при их полном отсутствии. Или если в компании не поддерживается актуальность документации

Повторное использование документации

Повторное использование документации применяется в том случае, если есть уже один или несколько подобных завершённых проектов.

Техническое задание, подготовленное на предыдущем проекте, может быть использовано для другого проекта с целью сократить продолжительность сбора, анализа и разработки требований, что позволит быстрее начать разработку. Например, ТЗ для интернет-магазинов похожи друг на друга и содержат одинаковые требования.

В большинстве случаев для нового проекта актуальна только часть документации, поэтому потребуется тщательная проверка требований на соответствие текущим целям и задачам заказчика.

Преимущества	Недостатки
Сокращение времени на разработку документации	1. Высокая стоимость первого проекта. 2. Излишняя детализация требований может привести к их дорогостоящим изменениям в будущем

Представитель заказчика в компании разработчика

Представитель заказчика в компании разработчика — это один из наиболее эффективных методов сбора требований. Он позволяет в короткие сроки получать от представителя заказчика своевременную оценку прогресса и корректности реализации, обратную связь (фидбэк) и дополнительную информацию для корректировки и разработки требований.

Метод часто применяется для сбора требований и управления ими при итерационной разработке, позволяет оперативно собирать, согласовывать и дорабатывать требования. Наличие представителя заказчика в компании разработчика — одно из главных правил Agile.

Преимущества	Недостатки
Быстрое получение обратной связи и информации от заказчика	1. Достаточно высокая цена для заказчика. 2. Временные затраты на адаптацию сотрудника

Работа в поле

Работа в поле — это наблюдение за тем, как работают пользователи, и документирование процесса, задач и результатов их деятельности.

Метод позволяет избежать проблем, связанных с трудностями стейкхолдеров в описании и выражении своих потребностей. В некоторых случаях процесс наблюдения может сопровождаться интервьюированием пользователей для уточнения особенностей и деталей их работы и задач. В процессе наблюдения можно также выявить пути оптимизации бизнес-процессов заказчика.

Преимущества	Недостатки
1. Позволяет наглядно увидеть проблему и разработать наиболее оптимальный вариант её решения. 2. Помогает наиболее точно собрать требования, наблюдая за работой сотрудников	1. В процессе наблюдения могут быть упущены некоторые альтернативные сценарии бизнес-процесса. 2. Трудно применить на секретных предприятиях или опасных (вредных) производствах

Обучение

Обучение — это процесс, в котором заказчик или любой другой человек из организации заказчика, знающий процесс, обучает аналитика по принципу «учитель – ученик».

Метод полезен, когда процессы и деятельность сотрудников заказчика трудно описать с помощью других методов или заказчик не может предоставить адекватное описание требований.

Обучение позволяет лучше понять сложные бизнес-процессы, а также преодолеть трудности, связанные с нехваткой абстрактного мышления и самовыражения у будущих пользователей системы.

Преимущества	Недостатки
1. Позволяет понять сложный бизнес процесс и предложить наилучшее решение	1. Высокая стоимость и длительность. 2. Метод неприменим на опасных (вредных) производствах

Мозговой штурм

Мозговой штурм — наиболее часто используемый метод получения требований, которые связаны с новыми или плохо изученными направлениями деятельности организации заказчика или функциями системы.

Он позволяет собрать множество идей от различных заинтересованных лиц (стейкхолдеров) в кратчайшие сроки и практически бесплатно. Во время мозгового штурма участники «накидывают» любые идеи, касающиеся решения данной проблемы.

С помощью этой методики можно проработать несколько различных вариантов решения заданной проблемы, а также разрешить конфликты требований.

Преимущества	Недостатки
1. Позволяет генерировать множество вариантов решений, в том числе и нестандартных. 2. Участники могут развивать идеи друг друга	1. Участники мозгового штурма должны быть мотивированы. 2. Метод трудно применим в распределённых командах

Совещание

Совещание — встреча, ориентированная на обсуждение конкретных вопросов, которые были определены и озвучены участникам заранее.

На такие встречи привлекаются люди, которые придерживаются различных точек зрения по текущей проблеме и могут помочь описать требования, основываясь на взглядах с разных сторон. В процессе совещания уточняется общий список требований, выявляются скрытые требования и решаются конфликты требований.

Совещания — одна из ключевых практик в Agile, так как в них участвуют все стороны, заинтересованные в развитии проекта и решении проблемы.

Преимущества	Недостатки
Позволяет развить и детализировать требования, определить приоритеты	<ol style="list-style-type: none"> 1. Если команда географически разделена, могут возникнуть трудности с присутствием всех необходимых людей на совещании. 2. Консенсус не обязательно будет достигнут

Use cases

Use cases (варианты использования) позволяют собрать и сформировать функциональные требования от лица участников. Диаграммы вариантов использования определяют границы решения и показывают связи с внешними системами и участниками.

Метод позволяет детализировать требования с точки зрения пользователей, а также уточнить и систематизировать функционал, который требуется реализовать.

Преимущества	Недостатки
Позволяет проработать все варианты развития сценария: основной и альтернативные сценарии	<ol style="list-style-type: none"> 1. Консенсус не обязательно будет достигнут. 2. Метод не применим для сбора нефункциональных требований

Этапы разработки требований

Выявление требований — самый трудный, важный, подверженный ошибкам и требующий интенсивного общения этап разработки ПО. С другой стороны, это творческий процесс и каждый использует свои инструменты для получения результата, то есть требований.

Стадию создания или разработки требований условно можно разделить на четыре этапа:

1. Выявление требований (сбор информации).
2. Анализ требований.
3. Спецификация (документация) требований.
4. Проверка требований.



Не стоит рассчитывать, что все действия по выявлению, анализу, спецификации и проверке требований будут выполнены последовательно. На практике процедура создания требований — это процесс итерационный:

1. Работая с пользователями, вы будете задавать вопросы, выслушивать ответы и наблюдать за их действиями — это **выявление** требований.
2. Затем вы обработаете полученную информацию, классифицируете по различным категориям и соотнесёте потребности клиентов с возможными требованиями к ПО — это **анализ**.
3. Потом вы оформите информацию от клиентов и выработанные требования в виде письменных документов и диаграмм — это **спецификация**.
4. В итоге предложите представителям пользователей подтвердить, что написанный вами текст точен и полон, и попросите их исправить возможные ошибки — это **проверка** требований.

Данные этапы будут выполняться, чередуясь и периодически повторяясь до тех пор, пока требования не будут собраны полностью.

Определение и описание методологических и технических требований — шаги, которые во многом определяют успех всего проекта, поскольку именно они влияют на все остальные этапы.

Выявление требований

Основные цели выявления требований в проектах: получение максимума информации о заказчике и специфике его задач, уточнение рамок проекта, оценка возможных рисков, а также формирование проектной группы, на которую будет возложена значительная часть предстоящих работ. Существуют различные приёмы для создания требований.

Определение процесса формулирования требований. Документирование этапов выявления, анализа, определения и проверки требований, наличие инструкций по выполнению ключевых операций — всё это поможет аналитикам качественно и согласованно выполнить их работу. Кроме того, будет проще поставить задачи по созданию требований и графики, а также продумать обеспечение необходимыми ресурсами.

Определение образа и границы проекта. Документ об образе и границах проекта содержит бизнес-требования к продукту. Описание образа проекта позволит всем заинтересованным лицам в общих чертах понять назначение продукта. Границы проекта определяют, что следует реализовать в этой версии, а что — в следующих. Образ и границы проекта — хорошая база для оценки предлагаемых требований. Образ продукта должен оставаться от версии к версии относительно стабильным, но для каждого выпуска необходимо составлять отдельный документ о границах.

Определение классов пользователей и их характеристик. Чтобы не упустить из виду потребности отдельных пользователей, их необходимо сгруппировать. Например, по частоте работе с ПО, используемым функциям, уровню привилегий и навыкам работы. Далее следует описать их обязанности, местоположение и личные характеристики, способные повлиять на архитектуру продукта.

Выбор в каждом классе пользователей. Сторонник продукта (product champion) — это человек, который сможет точно передавать настроения и нужды клиентов. Он представляет потребности определённого класса пользователей и принимает решения от их лица. В случае разработки внутрикорпоративных информационных систем такого человека выбрать проще. При коммерческой разработке можно расспросить клиентов или использовать площадки бета-тестирования. Выбранные люди должны принимать постоянное участие в проекте и обладать полномочиями для принятия решений, касающихся пользовательских требований.

Создание фокус-групп типичных пользователей. Необходимо выделить группы типичных пользователей предыдущих версий разрабатываемого продукта или похожих. Выяснить у них подробности о функциональности и качественных характеристиках разрабатываемого продукта. Фокус-группы особенно значимы при разработке коммерческих продуктов, когда приходится иметь дело с большой и разнородной клиентской базой. В отличие от сторонников продукта, у фокус-групп обычно нет полномочий на принятие решений.

Работа с пользователями для выяснения назначения продукта. Для этого следует выяснить у пользователей, какие задачи им требуется выполнять средствами ПО. Узнать, как должен клиент

взаимодействовать с системой для выполнения каждой такой задачи. Здесь удобно применить стандартный шаблон для документирования всех задач и для каждой сформулировать функциональные требования. Похожий способ часто применяется в правительственных проектах — создать документ с концепциями операций (ConOps), где указаны характеристики новой системы с точки зрения пользователя (IEEE, 1998a).

Определение системных событий и реакции на них. Определите возможные внешние события и ожидаемую реакцию системы на них. Это могут быть сигналы и данные, получаемые от внешнего оборудования, а также временные события, вызывающие ответную реакцию, например ежевечерняя передача данных, генерируемых системой, внешнему объекту. В бизнес-приложениях бизнес-события напрямую связаны с задачами.

Наблюдение за пользователями на рабочих местах. Наблюдая за работой пользователей, выявляют контекст потенциального применения нового продукта (Beyer и Holtzblatt, 1998). Простые диаграммы рабочих потоков, а также диаграммы потоков данных позволяют выяснить, где, как и какие данные задействовал пользователь. Документируя ход бизнес-процесса, удаётся определить требования к системе, предназначенной для поддержки этого процесса. Иногда даже выясняется, что для выполнения деловых задач клиентам вовсе и не требуется новое ПО (McGraw и Harbison, 1997).

Проведение совместных семинаров. Совместные семинары по выявлению требований, где тесно сотрудничают аналитики и клиенты, — отличный способ выявить нужды пользователей и составить наброски документов с требованиями (Gottesdiener, 2002). Конкретные примеры таких семинаров: Joint Requirements Planning (JRP) — совместное планирование требований (Martin, 1991) и Joint Application Development (JAD) — совместная разработка приложений (Wood и Silver, 1995).

Изучение отчётов о проблемах работающих систем с целью поиска новых идей. Поступающие от клиентов отчёты о проблемах и предложения о расширении функциональности — отличный источник идей о возможностях, которые можно реализовать в следующей версии или новом продукте. За подобной информацией стоит обратиться и к персоналу службы поддержки.

Повторное использование требований в разных проектах. Если необходимая клиенту функциональность аналогична уже реализованной в другом продукте, подумайте, готовы ли клиенты гибко пересмотреть свои требования для использования существующих компонентов. Требования, соответствующие бизнес-правилам компании, можно применить в нескольких проектах. Это требования к безопасности, определяющие порядок доступа к приложениям, и требования, соответствующие постановлениям правительства.

Анализ требований

Цель анализа требований в проектах — получить максимум информации о заказчике и специфике его задач, уточнить рамки проекта, оценить возможные риски, а также сформировать проектную группу, на которую будет возложена значительная часть предстоящих работ.

На этом этапе происходит идентификация принципиальных требований методологического и технологического характера, формулируются цели и задачи проекта, а также определяются критические факторы успеха, которые впоследствии будут использоваться для оценки результатов внедрения. Анализ требований выполняется на основе совещаний и собеседований с руководителями и специалистами заказчика, а продолжительность этого этапа, в зависимости от сложности задач и масштаба внедрения, может составлять от нескольких дней до нескольких недель.

Создание контекстной диаграммы. Контекстная диаграмма — простая модель анализа, которая отображает место новой системы в соответствующей среде. Она определяет границы и интерфейсы между разрабатываемой системой и сущностями, внешними для этой системы, например пользователями, устройствами и прочими информационными системами.

Создание пользовательского интерфейса и технических прототипов. Если разработчики или пользователи не совсем уверены насчёт требований, необходимо создать прототип — частичную, возможную или предварительную версию продукта, которая сделает концепции и возможности более осязаемыми. Оценка прототипа поможет всем заинтересованным лицам достичь взаимопонимания по решаемой проблеме.

Анализ осуществимости требований. Стоит проанализировать, насколько реально выполнить каждое требование при разумных затратах и с приемлемой производительностью в предполагаемой среде. Рассмотреть риски, связанные с реализацией каждого требования, включая конфликты с другими требованиями, зависимость от внешних факторов и препятствия технического характера.

Определение приоритетов требований. Аналитический подход помогает расставить приоритеты реализации функций продукта, решаемых задач или отдельных требований. На основании приоритетов определяется, в какой версии будет реализована та или иная функция или набор требований. Изменения распределяются по конкретным версиям, и затраты, необходимые на внесение изменений, включаются в план выпуска этих версий. В ходе работы над проектом приоритеты корректируются в соответствии с потребностями клиента, условиями рынка и бизнес-целями.

Моделирование требований. В отличие от спецификации требований к ПО или пользовательского интерфейса прототипа, где представлена подробная информация, графическая модель анализа отображает требования на высоком уровне абстракции. Модели позволяют выявить некорректные, несогласованные, отсутствующие и избыточные требования. К таким моделям относятся диаграммы потоков данных, диаграммы «сущность – связь», диаграммы перехода состояний, называемые также автоматами (statecharts), карты диалогов, диаграммы классов, диаграммы последовательностей, диаграммы взаимодействий, таблицы решений и деревья решений.

Создание словаря терминов. В нём собирают определения всех элементов и структур данных, связанных с системой, что позволяет всем участникам проекта использовать согласованные определения данных. На стадии работы над требованиями словарь должен содержать определения

элементов данных, относящихся к предметной области, чтобы клиентам и разработчикам было проще общаться.

Распределение требований по подсистемам. Требования к сложному продукту, включающему несколько подсистем, следует соразмерно распределять между программными, аппаратными и операторскими подсистемами и компонентами (Nelsen, 1990). Как правило, это делает системный инженер или разработчик.

Применение технологий развёртывания функций качества. Технология развёртывания функций качества Quality Function Deployment, QFD — точная методика, соотносящая возможности и атрибуты продукта с их значимостью для клиента (Zultner, 1993; Pardee, 1996). Она позволяет аналитически выявить функции, которые максимально удовлетворят потребности клиента.

Технология рассчитана на три класса требований:

- ожидаемые, которые клиент может не упомянуть, но будет расстроен, если их не окажется в продукте;
- обычные требования;
- отдельные, специальные требования, которые обеспечивают удобство работы клиентам, но отсутствие которых не влечёт санкций со стороны клиента.

Спецификации требований

Использование шаблона спецификации требований к ПО. Стандартный шаблон для документирования требований к ПО предоставляет согласованную структуру, позволяющую фиксировать описания нужной функциональности, а также прочую информацию, касающуюся требований. Многие компании начинают с использования шаблона спецификации требований к ПО, описанного в стандарте IEEE 830-1998 (IEEE, 1998b). Если компания занимается разными проектами, например проектирует новое крупное приложение и параллельно дорабатывает версии старых программ, необходимо создать соответствующие шаблоны для всех типов проектов. Шаблоны и процессы должны быть масштабируемыми.

Определение источников требований. Чтобы гарантировать, что все заинтересованные лица понимают, почему то или иное требование зафиксировано в спецификации требований к ПО, и упростить последующее прояснение требований, должны быть выявлены источники всех требований. Это может быть вариант использования или другая информация от пользователей, системное требование высокого уровня, бизнес-правило или иной внешний фактор. После указания всех лиц, заинтересованных в каждом требовании, будет понятно, к кому обратиться при поступлении запроса на изменение.

Присвоение уникальных идентификаторов всем требованиям. Необходимо разработать соглашение о присвоении уникальных идентификаторов требованиям, зафиксированным в

спецификации требований к ПО. Соглашение должно быть устойчивым к дополнению, удалению элементов и изменениям, вносимым в требования. Присвоение идентификаторов позволяет отслеживать требования и фиксировать вносимые изменения.

Указание атрибутов качества. Выявляя качественные характеристики, удовлетворяющие потребности клиента, не следует ограничиваться только обсуждением функциональности. Необходимо выяснить ожидаемые производительность, эффективность, надёжность, удобство использования и так далее. Информация от клиентов об относительной важности тех или иных качественных характеристиках позволит разработчику принять правильные решения, касающиеся архитектуры приложения.

Документирование бизнес-правил. К бизнес-правилам относятся корпоративные политики, правительственные распоряжения и алгоритмы вычислений. Список бизнес-правил ведётся отдельно от спецификации требований к ПО, поскольку правила обычно существуют вне рамок конкретного проекта. Для выполнения некоторых правил приходится создавать реализующие их функциональные требования и определять связь между этими требованиями и соответствующими правилами.

Проверка требований

Изучение документов с требованиями. Официальная проверка документирования требований — один из наиболее ценных способов проверки качества ПО. Для этого собирают небольшую команду, члены которой представляют различные направления, например это могут быть аналитик, клиент, разработчик и специалист по тестированию. Они тщательно изучают спецификацию требований к ПО, модель анализа и соответствующую информацию на предмет недостатков. Также полезно провести в ходе формулирования требований их неофициальный предварительный просмотр. И хотя реализовать это на практике непросто, данный приём — один из самых ценных.

Тестирование требований. На основе пользовательских требований создают сценарии функционального тестирования и документируют ожидаемое поведение продукта в конкретных условиях. Сценарии тестирования изучают совместно с клиентами и проверяют, что они отражают нужное поведение системы. Далее прослеживают связь сценариев тестирования с функциональными требованиями, чтобы удостовериться, что ни одно требование не пропущено и для всех требований есть соответствующие сценарии тестирования.

Определение критериев приемлемости. Пользователям предлагают описать, как они собираются определять соответствие продукта их потребностям и его пригодность к работе. Тесты на приемлемость следует основывать на сценариях использования (Hsia, Kung и Sell, 1997).

Управление требованиями

Цель управления требованиями состоит в том, чтобы гарантировать, что организация документирует, проверяет и удовлетворяет потребности и ожидания её клиентов и внутренних или внешних заинтересованных лиц. Управление требованиями начинается с выявления и анализа целей и ограничений клиента. Далее оно включает поддержку требований, интеграцию требований и организацию работы с требованиями и сопутствующей информацией, поставляющейся вместе с требованиями.

Управление требованиями включает общение между проектной командой и заинтересованными лицами с целью корректировки требований на протяжении всего проекта. Постоянное общение всех участников проекта важно, чтобы ни один класс требований не доминировал над другими.

1. Определение процесса управления изменениями.

Процесс представления, анализа и утверждения или отклонения изменений должен быть определён и применяться для управления всеми предлагаемыми изменениями.

2. Создание совета по управлению изменениями.

Из представителей заинтересованных в проекте лиц организуют совет по управлению изменениями, который будет получать информацию о предполагаемых изменениях требований, оценивать её, решать, какие изменения принять, а какие отклонить, и определять, в какой версии продукта будет внедрена та или иная функция.

3. Анализ влияния изменений требований.

Подобный анализ помогает совету по управлению изменениями принимать обоснованные решения. Следует оценивать, как каждое предлагаемое изменение требований повлияет на проект. Далее на основе матрицы связей выявляются другие требования, элементы архитектуры, исходный код и сценарии тестирования, которые, возможно, придётся изменить. В итоге определяется, что необходимо для реализации изменений, и оцениваются затраты на реализацию.

4. Создание базовой версии и управление версиями требований.

Базовая версия содержит требования, утверждённые для реализации в конкретной версии продукта. После определения базовых требований изменения можно вносить только в соответствии с процессом управления изменениями. Всем версиям спецификации требований присваиваются уникальные идентификаторы, чтобы закрепить версию. Более надёжное

решение — управлять версиями документов с требованиями при помощи соответствующих средств управления конфигурацией.

5. Ведение журнала изменений требований.

Должны быть зафиксированы даты изменения спецификаций требований, а также сами коррективы, их причины и лица, вносившие изменения. Для автоматизации используется утилита управления версиями или утилита управления требованиями.

6. Контроль за состоянием всех требований.

Создаётся БД, включающая по одной записи для каждого дискретного функционального требования. В БД заносятся ключевые атрибуты каждого требования, включая его состояние, например «предложено», «одобрено», «реализовано» или «проверено». Это нужно, чтобы в любой момент вы могли узнать количество требований в каждом состоянии.

7. Оценка изменяемости требований.

Еженедельно фиксируют количество требований, внесённых в базовую версию, а также число предложенных и одобренных изменений: добавлений, модификаций и удалений. Если требования формируются не самим клиентом, а от его лица, может оказаться, что проблема понята плохо, границы проекта определены нечётко, бизнес стремительно меняется, при сборе информации многие требования были упущены или внутрикорпоративные политики меняются в худшую сторону.

8. Использование средств управления требованиями.

Коммерческие утилиты управления требованиями позволяют хранить различные типы требований в БД. Для каждого требования можно определить атрибуты, отслеживать его состояние, а также выявить связи между требованиями и другими рабочими продуктами.

9. Создание матрицы связей требований.

Это таблица, сопоставляющая все функциональные требования с элементами архитектуры и кода, которые реализуют данное требование, и с тестами, проверяющими его. Матрица связей требований позволяет также сопоставить функциональные требования с требованиями более высоких уровней, на основе которых они созданы, и с другими родственными требованиями. Эту матрицу заполняют на входе, а не в конце работы над проектом.

Управление требованиями в рамках управления проектом

Выбор цикла разработки ПО. Следует определить несколько жизненных циклов разработки для проектов различного типа и различных степеней неопределённости требований. Менеджер проекта должен выбрать и использовать цикл, оптимальным образом подходящий для его проекта, включающий цикл операции по созданию требований. Если на ранних этапах работы над проектом требования или границы проекта определены нечётко, продукт следует разрабатывать постепенно,

небольшими этапами, начиная с наиболее понятных требований и устойчивых элементов архитектуры. Периодически выпускают промежуточные версии продукта, чтобы как можно раньше предоставлять клиенту работоспособные образцы приложения.

Планы реализации проекта должны быть основаны на требованиях. Планы и графики работы над проектом разрабатывают постепенно, по мере прояснения границ и подробных требований. Начинают с оценки затрат, которые необходимы для реализации функциональных требований, определённых на основе первоначального образа и границ продукта. Графики и оценка затрат, построенные на основе нечётких требований, окажутся крайне неточными, однако по мере детализации требований их следует уточнить.

Пересмотр обязательств по проекту при изменении требований. Добавляя в проект новые требования, необходимо оценивать, удаётся ли соблюдать обязательства, касающиеся графика и требований к качеству, при доступном объёме ресурсов. Если нет, то стоит обсудить реалии проекта с менеджерами и согласовать новые, достижимые обязательства (Humphrey, 1997; Fisher, Ury, и Patton, 1991; Wiegers, 2002). Либо пересмотреть планы по реализации проекта.

Документирование и управление рисками, связанными с требованиями. Одна из составляющих управления рисками проекта — выявление и документирование рисков, связанных с требованиями. Уменьшайте или предотвращайте их посредством мозговых штурмов, реализуйте корректирующие действия и отслеживайте их эффективность.

Контроль объёма работ по созданию требований. Усилия, прилагаемые командой для разработки требований и управления проектом, должны быть зафиксированы. Эти данные позволяют оценить соответствие планам и эффективнее спланировать необходимые ресурсы для будущих проектов. Также отслеживайте, как ваши действия по регламентации требований влияют на проект в целом. Это позволит оценить отдачу от работы.

Извлечение уроков из полученного опыта. Для этого в организации следует провести ретроспективу проектов, называемую также изучением законченных проектов (Robertson и Robertson, 1999; Kerth, 2001; Wiegers и Rothman, 2001). Ознакомление с опытом в области проблем и способов создания требований, которые накопились в ходе работы над предыдущими проектами, помогает менеджерам и аналитикам требований более эффективно работать в будущем.

Практическое задание

1. Возьмите список бизнес-требований из домашнего задания №2. В соответствии с приведённым критерием отличия бизнес-требований от бизнес-правил напишите три бизнес-правила.

2. Выберите три метода, которые вы планируете использовать для выявления требований для реализации своего бизнес-процесса, и кратко опишите для каждого, как вы будете это делать.
3. Обозначьте преимущества и недостатки каждого метода, после чего в трёх предложениях обоснуйте, почему именно эти методы подойдут для вашего процесса лучше всего.

Глоссарий

Use cases (варианты использования) позволяют собрать и сформировать функциональные требования от лица участников.

Автозапись — работа с записями, письмами (электронными письмами), а также с любыми другими документами, автор которых заказчик или конечный пользователь, то есть стейкхолдер.

Анкетирование — составление листа-опросника (анкеты, брифа), который может содержать как открытые вопросы, когда требуется сформулировать ответ, так и закрытые, когда требуется выбрать ответ из предложенных вариантов.

Бизнес-правило — это конкретная, исполняемая и проверяемая директива, которая находится под контролем организации и служит для поддержания бизнес-политики. Бизнес-правила представляют собой сочетание принципов и выводов, которые определяют, как мы делаем бизнес.

Бизнес-требования — это набор определённых требований по отношению к конкретному проекту, процессу или информационной системе, которые обычно выдвигает сам заказчик проекта или его ключевые пользователи.

Бизнес-цель — это результаты, которых владелец бизнеса стремится достичь и которые важно сохранить при управлении.

Изучение документации используется при наличии в организации документации, которая может помочь в определении потребностей заказчика.

Интервью — своего рода беседа по душам, тет-а-тет с заинтересованным лицом.

Мозговой штурм — наиболее часто используемый метод получения требований, которые связаны с новыми или плохо изученными направлениями деятельности организации заказчика или функциями системы.

Наличие представителя заказчика в компании разработчика — получение от представителя заказчика в короткие сроки обратной связи (фидбэка) и дополнительной информации для корректировки и разработки требований.

Обучение — процесс, в котором заказчик или любой другой человек из организации заказчика, знающий процесс, обучает аналитика по принципу «учитель – ученик».

Повторное использование документации применяется в том случае, если есть уже один или несколько подобных завершённых проектов.

Работа в поле — наблюдение за тем, как работают пользователи, и документирование процесса, задач и результатов их деятельности.

Совещание — встреча, ориентированная на обсуждение конкретных вопросов, которые были определены и озвучены участникам заранее.

Требование — это любое условие, которому должна соответствовать разрабатываемая система или программное средство. Требованием может быть возможность, которой система должна обладать, и ограничение, которому система должна удовлетворять.

Требования — это спецификация того, что должно быть реализовано. В них описано поведение системы, свойства системы или её атрибуты. Они могут быть ограничены процессом разработки системы.

Используемые источники

1. [Виды бизнес-правил.](#)
2. [Документирование бизнес-правил.](#)
3. [Анализ требований по Вигерсу.](#)
4. [Методы сбора требований или «Как понять, что хочет заказчик?»](#)
5. [Как стать настоящим аналитиком? Выявляем требования.](#)