

Хорошие приемы создания требований

Предмет: Управление требованиями и проектирование ПО

1. Обучение

2. Разработка требований

- Выявление требований
- Анализ выявленных требований
- Спецификация требований
- Проверка требований

3. Управление требованиями

4. Управление проектом

1. ОБУЧЕНИЕ

Не все разработчики умеют формулировать требования к ПО и общаться с клиентом. Обучение позволяет повысить профессиональные навыки сотрудников, выполняющих роли аналитиков, но не может компенсировать нехватку навыков межличностного общения и отсутствие интереса к делу.

Все участники проекта должны понимать концепцию и приемы формулирования требований. Эффективный способ **создать команду** — собрать участников проекта на однодневный семинар, где и обсудить все требования.

- Обучение аналитиков требований
- Ознакомление пользователей и менеджеров с требованиями
- Ознакомление разработчиков с предметной областью
- Создание бизнес-словаря

Обучение аналитиков требований.

Всем членам команды, которые будут исполнять функции аналитиков, необходимо научиться приемам формулирования требований — это может занять несколько дней.

Квалифицированный аналитик требований терпелив и методичен, обладает навыками межличностного общения и коммуникативными навыками, сведущ в предметной области и знает множество способов формулирования требований к ПО.

Ознакомление пользователей и менеджеров с требованиями.

Пользователи, которые будут принимать участие в разработке ПО, должны пройти непродолжительный тренинг (один-два дня), чтоб научиться формулировать требования. Он полезен и для менеджеров по разработке и по работе с клиентами. Обучение поможет понять особое значение выделения требований, суть процесса их разработки, а также опасность пренебрежения ими.

Ознакомление разработчиков с концепциями предметной области.

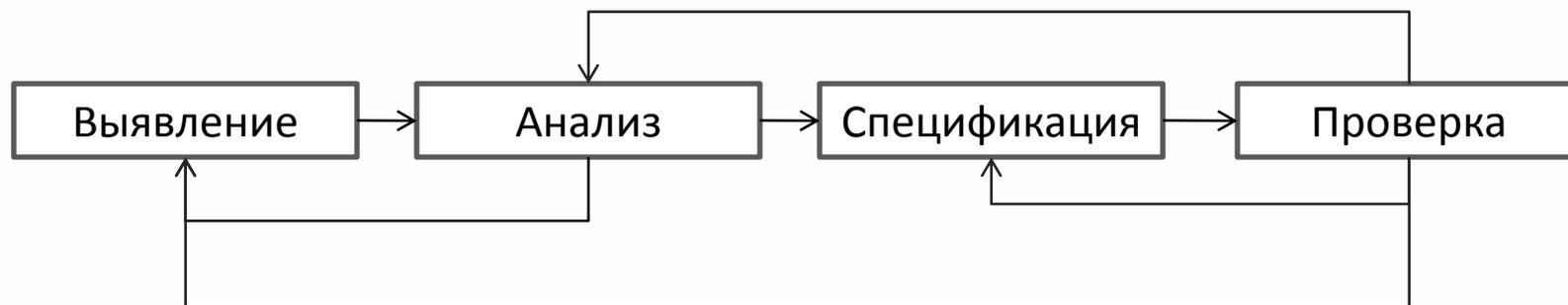
Чтобы помочь разработчикам в общих чертах понять предметную область, проведите семинар, на котором познакомьте их с бизнесом клиента, терминологией и назначением создаваемого продукта.

Это уменьшит вероятность путаницы, непонимания и доработок. Можно также на время проекта назначить каждому разработчику «личного пользователя», который будет разъяснять профессиональные термины и бизнес-концепции. Лучше, если это будет настоящий фанат продукта.

Создание бизнес-словаря.

Словарь со специализированными терминами из предметной области снизит вероятность непонимания. Включите в него синонимы, термины, имеющие несколько значений, и термины, имеющие в предметной области и повседневной жизни разные значения.

Разработка требований



Все действия выполняются попеременно, **поэтапно и повторяются**. Этот итерационный процесс или есть процедура создания требований.

Из-за разнообразия проектов по разработке ПО и организационных культур единого, **шаблонного подхода** к созданию требований **не существует**.

Оценив доступные вам способы общения с представителями пользователей, **выберите подходящие приемы** выявления требований (круглые столы, опросы и т.д.) и спланируйте время и ресурсы, необходимые для сбора информации.



Выявление требований

обсуждались три уровня требований: бизнес-уровень, пользовательский уровень и функциональный. Они собираются из разных источников на различных этапах работы над проектом, имеют различные цели и аудиторию и должны документироваться по-разному.

- Определение процесса формулирования требований
- Определение образа и границы проекта
- Определение классов пользователей
- Определение системных событий
- Проведение совместных семинаров
- Наблюдение за пользователями
- Изучение отчетов о проблемах
- Повторное использование



1. Определить процесс формулирования требований

- Задокументируйте этапы выявления, анализа, определения и проверки требований.
- Наличие инструкций по выполнению ключевых операций по может аналитикам качественно и согласованно выполнить их работу.



2. Определить 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 продукта должен оставаться **от версии к версии** относительно стабильным, но для каждого выпуска необходимо составлять отдельный документ о границах.



3. Определение классов пользователей и их характеристик.

Выделите пользователей в группы.

Например,

- по частоте работе с ПО, используемым функциям,
- уровню привилегий
- навыкам работы.

Опишите их **обязанности, местоположение и личные характеристики**, способные повлиять на архитектуру продукта.



3. Выбор сторонника продукта (product champion) в каждом классе пользователей.

- Это человек, который сможет точно **передавать настроения и нужды клиентов.**
- Он представляет **потребности определенного класса пользователей** и принимает решения от их лица.
- Выбранные вами люди должны принимать **постоянное участие в проекте и обладать полномочиями для принятия решений**, касающихся пользовательских требований.



4. Работа с пользователями для выяснения назначения продукта (бизнес-процессы).

- Выясните у пользователей, какие **задачи им требуется выполнять** средствами ПО.
- Обсудите, как должен **клиент взаимодействовать с системой** для выполнения каждой такой задачи.
- Воспользуйтесь **стандартным шаблоном** для документирования всех задач и для каждой сформулируйте функциональные требования.



5. 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.



6. 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.

Конкретные примеры таких семинаров — Joint Requirements Planning



7. Наблюдение за пользователями на рабочих местах.

Наблюдая за работой пользователей, **выявляют контекст потенциального** применения нового продукта.

Простые **диаграммы рабочих потоков**, а также диаграммы потоков данных позволяют выяснить, где, как и какие данные задействовал пользователь.

Документируя ход бизнес-процесса, удастся определить **требования к системе** предназначенной для поддержки этого процесса.



8. Изучение отчетов о проблемах работающих систем с целью поиска новых идей.

Поступающие от клиентов **отчеты 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**.



9. 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

Представляет собой

- детализацию требований, гарантирующую, что их понимают все заинтересованные лица.
- Исследование на предмет ошибок и недостатков.

Цель: достаточно качественно и подробно описать требования, позволяющие **оценить проект**

Включают:

- Создание контекстной диаграммы
- Создание прототипов
- Анализ осуществимости требований
- Определение приоритетов требований
- Моделирование требований
- Создание словаря терминов
- Распределение требований по подсистемам



Создание контекстной диаграммы

Контекстная диаграмма — простая модель анализа, отображающая 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 технических прототипов

Если разработчики или пользователи не совсем уверены насчет требований, **создайте 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

Проанализируйте, насколько **реально реализовать каждое требование** при разумных **затратах** и с приемлемой производительностью в предполагаемой среде.

Рассмотрите **риски**, связанные с реализацией каждого требования:

- конфликты с другими требованиями,
- зависимость от внешних факторов
- препятствия технического характера



Определение приоритетов требований

На основании приоритетов установите, **в какой версии будет реализована** та или иная функция или набор требований.

Подтверждая **изменения, распределите все их по конкретным версиям** и включите в план выпуска этих версий затраты, необходимые на внесение изменений.

В ходе работы над проектом периодически **корректируйте приоритеты** в соответствии с потребностями клиента, условиями рынка и бизнес-целями.



Моделирование требований

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,
- диаграммы классов,
- диаграммы последовательностей,
- диаграммы взаимодействий,
- таблицы решений и др.



Создание словаря терминов

Соберите **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
- операторскими подсистемами и компонентами.

Как правило, это осуществляет **системный инженер или разработчик.**



Спецификации требований

Документирование требования, чтобы это обеспечивало **удобный доступ и просмотр (в текстовом и графическом виде)**.

- Использование шаблона спецификации
- Определение источников требований
- Присвоение уникальных идентификаторов
- Указание атрибутов качества
- Документирование бизнес-правил



Использование шаблона спецификации требований к ПО

Шаблон **предоставляет согласованную структуру**, позволяющую фиксировать описания нужной функциональности, а также прочую информацию, касающуюся требований.

- **Создайте стандартный шаблон** для документирования требований к ПО в вашей организации.
- **Модифицируйте** один из существующих в соответствии со спецификой проекта.
- Многие компании начинают с **использования шаблона спецификации** требований к ПО, описанного в стандарте IEEE 830-1998



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, **выявите источники всех требований.**

Это может быть вариант использования или другая информация от пользователей, системное требование высокого уровня, бизнес-правило или иной внешний фактор.

Укажите всех лиц, заинтересованных в каждом требовании, вы будете знать, к кому обратиться при поступлении запроса на изменение.

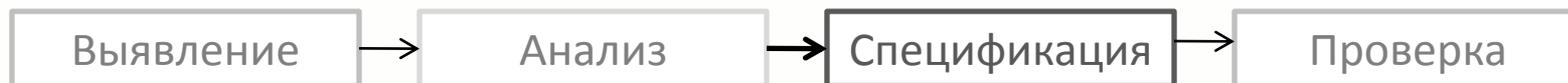


Присвоение уникальных идентификаторов всем требованиям

Выработайте **соглашение о присвоении уникальных идентификаторов** по 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.

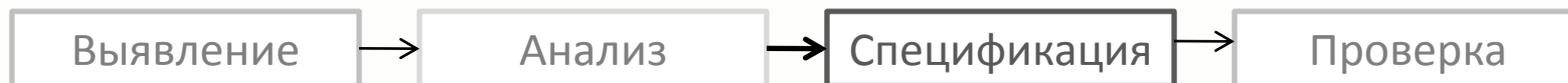


Указание атрибутов качества

Выясните ожидаемые:

- Производительность
- Эффективность
- Надежность
- удобство использования и др.

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.

Ведите список бизнес-правил отдельно от спецификации требований к ПО, поскольку правила обычно существуют вне рамок конкретного проекта.



Проверка требований

Проверка гарантирует, что все положения требований **корректны, отражают желаемые качественные характеристики** и удовлетворяют потребностям клиента.

Может оказаться, что требования, в спецификации требований к ПО выглядевшие превосходно, **при реализации чреватые проблемами.**

В большинстве случаев удастся выявить **двусмысленности и неопределенности**, написав для требований сценарии тестирования.

- Изучение документов с требованиями
- Тестирование требований



Изучение документов с требованиями

Соберите небольшую команду, члены которой представляют различные направления (например, аналитик, клиент, разработчик и специалист по тестированию), и **тщательно изучите спецификацию требований к ПО**, модель анализа и соответствующую информацию на предмет недостатков.

Также полезно провести в ходе формулирования требований их **неофициальный предварительный просмотр**.



Тестирование требований.

На основе пользовательских требований **создайте сценарии функционального тестирования** и задокументируйте ожидаемое поведение продукта в конкретных условиях.

Совместно с клиентами изучите сценарии тестирования и **убедитесь, что они отражают нужное поведение** системы.

Проследите связь сценариев тестирования с функциональными требованиями и удостоверьтесь, что **ни одно требование не пропущено** и что для всех требований есть соответствующие сценарии тестирования.

Управление требованиями

Начальные требования неизбежно **корректируются в процессе работы** клиентами, менеджерами, специалистами по маркетингу, разработчиками и другими лицами.

Для эффективного управления требованиями **необходим процесс**, позволяющий предлагать изменения и оценивать их возможную стоимость и влияние на проект.

- Определение процесса
- Создание совета по управлению изменениями
- Анализ влияния изменений требований
- Управление версиями требований
- Ведение журнала изменений
- Контроль за состоянием всех требований
- ПО для управления требованиями

Определение процесса управления изменениями

Определите:

процесс представления,
анализа и утверждения
отклонения изменений.

Применяйте его для управления всеми предлагаемыми изменениями

Создание совета по управлению изменениями

Из представителей заинтересованных в проекте лиц **организуите совет по управлению изменениями**, который:

- будет получать информацию о предполагаемых изменениях требований, оценивать ее,
- решать, какие изменения принять, а какие отклонить,
- определять, в какой версии продукта будет внедрена та или иная функция.

Анализ влияния изменений требований

Анализ влияния изменений помогает совету по управлению изменениями **принимать обоснованные решения.**

Оцените, как каждое предлагаемое изменение требований повлияет на проект.

На основе матрицы связей **выявите другие требования**, элементы архитектуры, исходный код и сценарии тестирования, которые, возможно, придется изменить.

Определите, что необходимо для реализации изменений, и оцените затраты на реализацию.

Управление версиями требований.

Присвойте всем версиям спецификации требований уникальные идентификаторы, чтобы избежать путаницы между черновыми вариантами и базовыми версиями, а также между предыдущей *и* текущей версиями требований.

Ведение журнала изменений требований

Фиксируйте даты изменения спецификаций требований, сами коррективы, их причины, а также лиц, вносивших изменения.

Контроль за состоянием всех требований

Создайте БД, включающую по одной записи для каждого дискретного функционального требования.

Занесите ключевые атрибуты каждого требования, включая его состояние {например «предложено», «одобрено», «реализовано» или «проверено»}, чтобы в любой момент вы могли узнать количество требований в каждом состоянии

Используйте ПО для управления требованиями

Управление проектом

Способы управления проектом ПО тесно связаны с работой над требованиями к нему.

Планируйте ресурсы, графики и обязательства по проекту на основании требований, которые собираетесь реализовать.

Изменения требований влияют на планы реализации, поэтому в планах следует **предусмотреть возможность частичного изменений требований** и расширения границ проекта.

Включает:

- Выбор цикла разработки ПО
- План реализации проекта
- Пересмотр обязательств при изменении требований
- Управление рисками

Выбор цикла разработки ПО

Вашей компании следует **определить несколько жизненных циклов** разработки для проектов различного ПО.

Если на ранних этапах работы над проектом требования или границы проекта определены нечетко, **разрабатывайте продукт постепенно**, начиная с наиболее понятных требований.

По возможности **реализуйте наборы функций**, чтобы периодически выпускать промежуточные версии продукта и как можно раньше предоставлять клиенту работоспособные образцы приложения

Планы реализации проекта

- Должны быть основаны на требованиях.
- Разрабатывайте планы и графики работы над проектом.
- Начните с оценки затрат, необходимых на реализацию функциональных требований, определенных на основе первоначального образа и границ продукт.

Пересмотр обязательств по проекту при изменении требований

Добавляя в проект новые требования, оцените, удастся ли соблюдать обязательства, касающиеся графика и требований к качеству, при доступном объеме ресурсов.

Если нет, **обсудите реалии проекта менеджерами** и согласуйте новые, достижимые обязательства.

Документирование и управление рисками

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.